ITWissen

Das große Online-Lexikon für Informationstechnologie

# Glossar

# Codierung

# Index

ITWissen
Das große Online-Lexikon für Informationstechnologie

# Index

ITWissen
Das große Online-Lexikon für Informationstechnologie

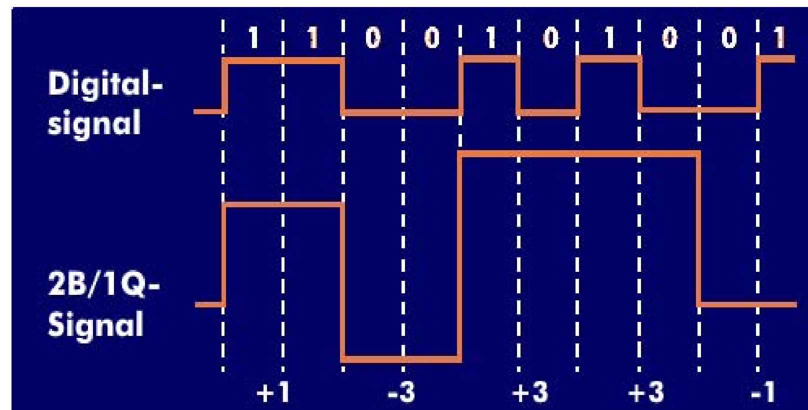
# Codierung

**1-aus-n-Code**
*1 out of n code*

Bei dem 1-aus-n-*Code* besteht jedes Maschinenwort aus einer „1". Alle restlichen Stellen sind „0". Bei diesem Code kommt es zu sehr umfangreichen Wortlängen. So könnte beispielsweise der Buchstabe „A" dem Maschinenwort 0001 entsprechen, der Buchstabe „B" dem 0010 und „C" dem 0100.
In der Version 1-aus-10 verschiebt sich das jeweilige Bit in dem zehnstelligen Datenwort vom Least Significant Bit (LSB) für den Dezimalwert 0: 0000000001, zum Most Significant Bit (MSB) für die Ziffer 9: 1000000000.

**2B1Q-Codierung**
*2B1Q, 2 binary 1 quaternary*

Bei der 2B1Q-*Codierung* (2 *Binary*, 1 Quaternary) werden zwei Bits, ein Dibit, in ein Vierpegel-Signal (quaternär) umgewandelt, wodurch die Schrittgeschwindigkeit auf der Leitung halbiert wird.
Bei der Umwandlung wird ein Zwei-Bit-Wort in ein Quarternärsymbol gewandelt, wobei das erste Bit als Vorzeichen, das zweite Bit (LSB) als Amplitude interpretiert wird. So wird die Binärkombination 10 zum Quarternärsymbol +3, die 11 zu +1, die 01 zu -1 und die 00 zu -3.




*2B/1Q-Codierung*

Das 2B1Q-Verfahren ist für High Bitrate Digital Subscriber Line (HDSL) und für das VDSL-Verfahren (Very High Speed Digital Subscriber Line) geeignet, wird aber auch bei ISDN genutzt. Eine weitere Reduzierung der Schrittgeschwindigkeit kann dadurch erreicht werden, dass der Datenstrom auf zwei Kanäle aufgeteilt wird.

ITWissen
Das große Online-Lexikon für Informationstechnologie

# Codierung

## 3-in-4-Codierung
*3 in 4 encoding*

Bei der 3-in-4-*Codierung* handelt es sich um ein zeichenorientiertes Codierverfahren. Mit diesem Verfahren können beliebig strukturierte Nutzdaten übertragen werden. Die Anwenderdaten werden dabei so codiert, dass ausschließlich Zeichen entstehen, die nicht für Steuernachrichten reserviert sind. Nachteilig ist der sehr große Overhead von 33% (Übertragungszeit, Datenvolumen), da die Nutzdaten jeweils in Blöcken zu 3 Bytes in Blöcke zu 4 Bytes des Kanalsignals umgesetzt werden. Anwendung findet die 3-in-4-Codierung beispielsweise zur transparenten Übertragung von Telesoftware in T-Online.

## 4B3T-Codierung
*4B3T, 4 binary, 3 ternary*

| Bitgruppe | Ternärwert |
|---|---|
| 0000 | 0 -1 +1 |
| 0001 | -1 +1 0 |
| 0010 | -1 0 +1 |
| 0011 | +1 -1 +1 |
| 0100 | 0 +1 +1 |
| 0101 | 0 +1 0 |
| 0110 | 0 0 +1 |
| 0111 | -1 +1 +1 |
| 1000 | 0 +1 -1 |
| 1001 | +1 -1 0 |
| 1010 | +1 0 -1 |
| 1011 | +1 0 0 |
| 1100 | +1 0 +1 |
| 1101 | +1 +1 0 |
| 1110 | +1 +1 -1 |
| 1111 | +1 +1 +1 |

*4B/3T-Codierung*

Die 4B3T-Codde ist ein Leitungscodierverfahren zur blockweisen Ternärcodierung von Binärsignalen. Bei 4B3T werden jeweils vier Bit (Binärblock 4B) mit drei *ternäre* Signalpegel (Ternärblock 3T) dargestellt. Das bedeutet, dass insgesamt 16 Binärwerte auf 27 Signalwerten abgebildet werden. Dieses *Codierverfahren* wird auch als MMS43-*Code* bezeichnet, was für Modified Monitored Sum (MMS) steht. Die 4B3T-Codierung arbeitet mit einer Tabelle, mit der die 16 Bitmuster zu je 4 Bit den ternären Signalen zugeordnet werden. Jedem 4-Bit-Wort werden vier unterschiedliche Ternärfolgen zugeordnet. Die Codierung wird als *Übertragungsverfahren* bei ISDN am Basisanschluss verwendet. Die 4B3T-Codierung wird im Anschlussbereich zwischen der ISDN-Vermittlungsstelle und dem Netzabschluss eingesetzt. Durch die ternäre Darstellung des Datenstromes wird die Datenrate von 160 kbit/s auf eine

# Codierung

Schrittgeschwindigkeit von 120 kBaud reduziert. Dadurch wird die maximale Entfernung von 8 km zwischen Netzwerkabschluss und ISDN-Vermittlungsstelle erreicht.

## 4B5B-Codierung

*4B5B, 4 binary, 5 binary*

Die 4B5B-*Codierung* wird in Hochgeschwindigkeitsnetzen benutzt. Bei dieser Codierung werden alle Daten in 4-Bit-Einheiten, ein so genantes Nibble oder ein *Quadbit*, unterteilt und nach einer Tabelle in 5-Bit-Einheiten (Symbole) umcodiert.
Diese Codiertabelle ist so aufgebaut, dass unabhängig von den Eingangsdaten nie Symbole mit mehr als drei Nullen in Folge auftreten. Der Vorteil dieser Codierung liegt darin, dass man die *NRZ-I-Codierung* nutzen kann, ohne dass bei langen Null-Sequenzen die Synchronisation verloren geht. Nachteilig ist der 25%ige Overhead, der durch die Umsetzung von 4-Bit- auf 5-Bit-Einheiten entsteht. Das bedeutet, dass bei FDDI, wo diese Codierung eingesetzt wird, die Datenrate auf dem Übertragungsmedium auf 125 Mbit/s erhöht werden muss, damit als Nutzdatenrate 100 Mbit/s zur Verfügung stehen. Von den 32 verschiedenen Zeichen, die mit dem 4B5B-*Code* erzeugt werden, werden 16 zur Nutzdatenübertragung benötigt, die restlichen 16 für Steuerzwecke. Bei der 4B5B-Codierung spricht man auch von einem »4 Bit Nibble to 5 Bit Symbol«.
Die von Ethernet her bekannte *Manchester-Codierung* kann aus

| 4-Bit-Wert Nibble | 5-Bit-Wert Symbole | 4-Bit-Wert Nibble | 5-Bit-Wert Symbole |
|---|---|---|---|
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

*Codiertabelle der 4B/5B-Codierung*

wirtschaftlichen Gründen in HSLNAs nicht verwendet werden, da sie bei jedem übertragenen Bit zweimal den Status wechselt und damit die doppelte Bandbreite belegt.

**4T+-Codierung**
*4T+, 4 ternary plus*

4T+ ist eine spezielle *Codierung* um Fast-Ethernet über einfache Telefonverkabelungen zu übertragen. Dieses spezifizierte Verfahren hatte Ende der 80er Jahre den Vorteil, dass man Fast-Ethernet über einfache Telefonkabel der Kategorie 3 übertragen konnte, ohne eine neue Verkabelungsinfrastruktur mit TP-Kabeln höherer Kategorie verlegen zu müssen. Die klassischen Telefonkabel waren in allen Gebäuden und Büros verlegt.
Beim 4T+-Verfahren wird als Zugangsverfahren CSMA/*CD* benutzt, wobei vier UTP-Kabel der Kategorie 3 Verwendung finden. Das Verfahren, das in 100Base-T4 angewendet wird, benutzt jeweils drei Leitungspaare für die Datenübertragung und das vierte Paar für die Kollisionsmeldung. Bei diesem Verfahren wird die *8B6T-Codierung* verwendet.

**5B6B-Codierung**
*5B6B, 5 binary, 6 binary*

Das *Codierverfahren* 5B6B wird in 100Base-VG eingesetzt und stellt 5-Bit-Wörter über Codiertabellen in Form von 6-Bit-Symbolen dar. Von den 32 6-Bit-Symbolen sind 20 gleichstromneutral, d.h. sie haben eine identische Anzahl an Einsen und Nullen und werden daher auch als »balanced« bezeichnet.
Die verbleibenden zwölf 6-Bit-Symbole, die »unbalanced« genannt werden, werden in zwei Gruppen mit jeweils 12 Symbolen mit zwei Einsen und mit vier Einsen definiert. Diese Maßnahme ist erforderlich, um Unbalanced-Symbole gleichstromfrei übertragen zu können. Zu diesem Zweck wird bei der Übertragung von aufeinander folgenden Unbalanced-Symbolen alternierend jeweils ein Symbol aus der einen Gruppe (mit zwei Einsen) gefolgt von einem Symbol aus der anderen Gruppe (mit vier Einsen) übertragen. Durch diesen Vorgang erhält man eine ausgeglichene Bitfolge aus Einsen und Nullen und somit eine gleichstromfreie

# Codierung

| 5-Bit-Symbol | 6-Bit balan. | 6-Bit-Code-Pair unbalanced | | 5-Bit-Symbol | 6-Bit balan. | 6-Bit-Code-Pair unbalanced | |
|---|---|---|---|---|---|---|---|
| | | zwei 1 | vier 1 | | | zwei 1 | vier 1 |
| 00000 | | 001100 | 110011 | 10000 | | 000101 | 111010 |
| 00001 | 101100 | | | 10001 | 100101 | | |
| 00010 | | 100010 | 101110 | 10010 | | 001001 | 110110 |
| 00011 | 001101 | | | 10011 | 010110 | | |
| 00100 | | 001010 | 110101 | 10100 | 111000 | | |
| 00101 | 010101 | | | 10101 | | 011000 | 100111 |
| 00110 | 001110 | | | 10110 | 011001 | | |
| 00111 | 001011 | | | 10111 | | 100001 | 011110 |
| 01000 | 000111 | | | 11000 | 110001 | | |
| 01001 | 100011 | | | 11001 | 101010 | | |
| 01010 | 100110 | | | 11010 | | 010100 | 101011 |
| 01011 | | 000110 | 111001 | 11011 | 110100 | | |
| 01100 | | 101000 | 010111 | 11100 | 011100 | | |
| 01101 | 011010 | | | 11101 | 010011 | | |
| 01110 | | 100100 | 011011 | 11110 | | 010010 | 101101 |
| 01111 | 101001 | | | 11111 | 110010 | | |

*5B/6B-Codierung, Codetabelle*

Übertragung. Gegenüber der *4B5B-Codierung* wird durch dieses Verfahren die Schrittgeschwindigkeit erhöht.

**7-Bit-Code**
*7 bit code*

Beim 7-Bit-*Code*, auch bekannt als IA-5-Alphabet, handelt es sich um den ASCII-Zeichensatz zur Darstellung von insgesamt 128 alphanumerischen Zeichen, Ziffern, Sonder- und Steuerzeichen. Dieser Code wurde international als Referenz-Code unter ISO/IEC 646 standardisiert. Die deutsche Referenz wurde von DIN unter DIN 66003 genormt.

Beim 7-Bit-Code wird jedes Zeichen mit 7 Bit dargestellt. Die Zeichen sind in einer Codetabelle gegliedert und sind durch die Spalten- und Zeilennummer festgelegt. Die Nummerierung kann in binärer oder hexadezimaler Schreibweise sein.

Wenn der 7-Bit-Code in Bytes dargestellt wird, wird das fehlende achte Bit durch ein Prüfbit ergänzt.

# Codierung

**8-Bit-Code**
*8 bit code*

Der 8-Bit-*Code* basiert auf dem *7-Bit-Code*, der um Sprachgruppen erweitert wurde und sich an Weltregionen orientiert. So ist gibt es mit Latein 1 *Code-Elemente* für die westeuropäischen Sprachen, ebenso wie für Amerika, Australien und Afrika. Latein 2 erfüllt die Anforderungen an die osteuropäischen Sprachen. Des Weiteren gibt es spezielle Code-Elemente für die hebräische, kyrillische, arabische oder andere Sprachen.
Beim 8-Bit-Code ist das Byte die kleinste darstellbare Einheit. Mit diesem Coder können 256 Zeichen *binär* dargestellt werden.

**8B6T-Codierung**
*8B6T, 8 binary, 6 ternary*

Bei der 8B6T-*Codierung* spricht man von »8 bit Word to 6 *Ternary* Symbols«. Es handelt sich um eine physikalische Codierung, bei der 8-Bit-Wörter auf 6 dreiwertige Symbole (Ternary) abgebildet werden. Dreiwertige Symbole repräsentieren die Werte Minus, Null und Plus (-, 0, +). Bei der 8B6T-Codierung werden die Bitkombinationen direkt in drei elektrische Pegel umgesetzt und kann unmittelbar für die Übertragung genutzt werden.
Die 8B6T-Codierung findet ihre Anwendung z.B. in IEEE 802.3 100Base-T4.

**8B10B-Codierung**
*8B10B, 8 binary, 10 binary*

Bei der 8B10B-*Codierung* werden 8-Bit-Wörter auf 10-Bit-Wörtern abgebildet. Mit der Konvertierung wird verhindert, dass sich lange Folgen von Nullen oder Einsen bilden, die die Taktgenerierung und Datenregenerierung erschweren.
Da bei einer Umsetzung einer 8-Bit-Folge 256 Bitkombinationen entstehen, bei einer 10-Bit-Folge hingegen 1024 Bitkombinationen, nutzt man für die Abbildung der 8-Bit-Worte nur die Bitkombinationen, die viele Pegelwechsel aufweisen. Und zwar werden nur die Bitkombinationen berücksichtigt, die maximal fünf aufeinander folgende Nullen und fünf aufeinander folgende Einsen enthalten. Dadurch kann aus dem codierten Signal, das hinreichend viele Pegelwechsel aufweist, das für die Synchronisation benötigte Taktsignal

# Codierung

abgeleitet werden.
Da bei der Abbildung von 8-Bit-Folgen in 10-Bit-Folgen nur 256 Bitkombinationen für die 8-Bit-Folgen benötigt werden, verbleiben weitere 768 Bitkombinationen. Diese werden für Sonderzeichen und im Fibre Channel für die Verwaltung des Links benutzt.
Die 8B10B-Codierung findet ihre Anwendung z.B. in Fibre-Channel, Gigabit-Ethernet, 10-Gigabit-Ethernet, bei ESCON, in der StarFabric sowie bei den ATM-Übertragungsschnittstellen, und zwar bei der direkten Zellenübertragung.

**64B/66B-Codierung**
*64B/66B encoding*

Die 64B/66B-*Codierung* ist ein sehr effizientes Codierverfahren, das in 10-Gigabit-Ethernet, 40-Gigabit-Ethernet und 100-Gigabit-Ethernet eingesetzt wird. Bei diesem Verfahren werden nur zwei zusätzliche Bits für die Übertragung eines 64-Bit-Datenblocks benötigt. Reine Daten (D0 - D7) werden in diesem Verfahren mit »01« codiert, Kontrollbytes (C0 - C7) und ein Gemisch aus Kontroll- und Datenbytes erhalten die Codierung »10«. Unbenutzte Bits werden als »0« übertragen.

**128B/130B-Codierung**
*128b130b encoding*

Bei PCI-Express 3.0 wird die bisherige *8B10B-Codierung* von einer 128B/130B-*Codierung* abgelöst. Die 128B/130B-Codierung zeichnet sich durch einen geringen Overhead von nur 1,6 Prozent aus, wodurch der steigende Energiebedarf bei der Übertragungsrate von 8 GB/s pro Lane wesentlich geringer ausfällt, als der der 8B10B-Codierung, die einen Overhead von 20 % hat. Einhergehend mit der Verringerung des Leistungsbedarfs steigt allerdings die Komplexität der Logiken. Es werden daher deutlich mehr Logikgatter in einem FPGA benötigt.

**A-Law-Verfahren**
*A-law method*

Das A-Law-Verfahren ist ein Verfahren für die Dynamikkompression von Audiosignalen, das in der ITU-Empfehlung G.711 beschrieben ist. Die Dynamikkompression dient der Verbesserung

des Störspannungsabstands (SNR) bei gleichen Übertragungsbedingungen. Das Verfahren verwendet eine logarithmische Dynamikkennlinie, die besonders bei niedrigen Eingangspegeln eine hohe Dynamik aufweist und bei hohen Eingangspegeln eine sehr geringe. Dadurch wird das Rauschen bei geringen Pegeln, also bei leisen Tönen reduziert.
Das A-Law-Verfahren ist ein ITU-Standard, der bei der Konvertierung zwischen analogen und digitalen Signalen in PCM-Systemen eingesetzt wird.
Das A-Law-Verfahren wird hauptsächlich in Europa verwendet, in den USA dagegen ein geringfügig in den Quantisierungsstufen abweichendes Verfahren, das µ-Law-Verfahren.
Dieses Verfahren zeichnet sich durch eine Dynamikkennlinie aus, die im Niedrigpegelbereich noch steiler ist als die des A-Law-Verfahrens.

**ACM, adaptive coding and modulation**

Verschiedene Funktechniken verwenden Optimierungstechniken um möglichst viele Daten über einen Funkkanal zu übertragen. Werden die funktechnischen Gegebenheiten allerdings beeinträchtigt, dann steigt die Fehlerrate. Vor diesem Hintergrund passen Unicast- und Broadcasttechniken ihre *Codierung* und Modulation der tatsächlichen Übertragungsqualität an. Bei ungestörtem Übertragungskanal wird ein effizienteres *Codierschema* benutzt als bei beeinträchtigter Übertragung.
Ein Beispiel hierfür ist DVB-S2, bei dem die Empfangsqualität vom Empfänger über einen Rückkanal zum Sender übertragen wird und für die Anpassung des Modulationsverfahrens sorgt. Dieses Verfahren heißt Adaptive Coding and Modulation (ACM).
Mit *Constant Coding and Modulation* (CCM) und *Variable Coding and Modulation* (VCM) gibt es weitere Steuerungsverfahren für die Codierung und Modulation, die allerdings nicht mit Rückkanal arbeiten.

# Codierung

**Aiken-Code**
*aiken code*

| Dezimal | Aiken-Code | | | | Dualsystem | | | |
|---|---|---|---|---|---|---|---|---|
| Wertigkeit | 2 | 4 | 2 | 1 | 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

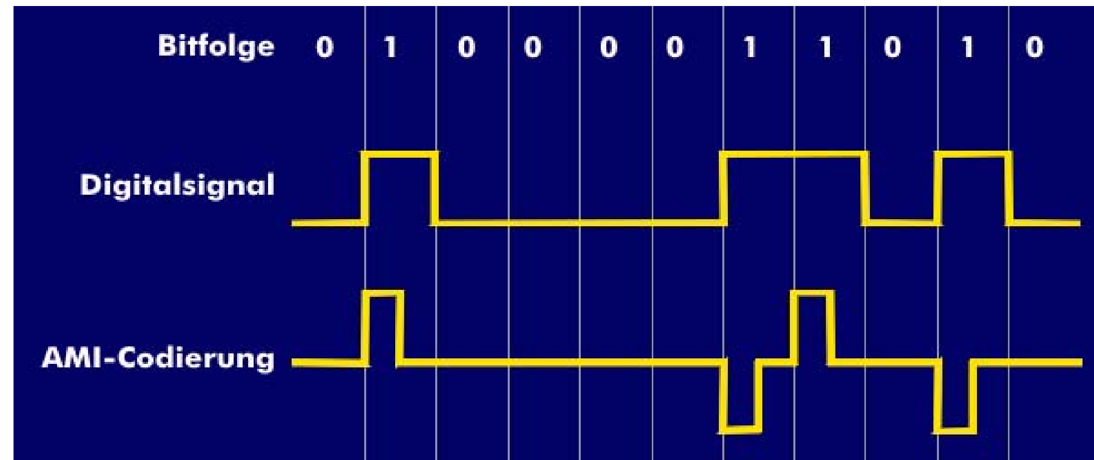
*Aiken-Code und Binärcode*

Der Aiken-*Code* ist ein vierstelliger *Binärcode*, allerdings mit einer anderen Wertigkeit als das Binärsystem. Beim Aiken-Code repräsentieren die vier Bits die Werte 2-4-2-1. Die ersten Dezimalziffern bis zur Dezimalzahl 4 entsprechen Aiken-codiert denen im Binärsystem, die Zahlen zwischen 5 und 9 entsprechen aber den dual dargestellten Zahlen zwischen 11 und 15. Jede Aiken-codierte Zahl kann in das Dualsystem übertragen werden. Dies nutzt man bei der Subtraktion von Zahlen, die durch einfache Bildung des 9er-Komplements und anschließender Addition durchgeführt werden kann.
Der Aiken-Code ist nach Howard Hathaway Aiken benannt, einem Professor für angewandte Mathematik an der Harvard University in Cambridge.

**AMI, alternate mark inversion**
*AMI-Codierung*

Die AMI-*Codierung* ist eines der grundsätzlichen Codierverfahren. Es handelt es sich um ein pseudoternäres Verfahren, das aus dem *NRZ*-Verfahren entwickelt wurde: zwei *binäre* Zustände werden durch drei Potenziale repräsentiert: 0, 1 und -1. Die binäre 0 hat wird als 0 codiert, die erste binäre 1 hat einen Pegel von +1 und die folgende binäre 1 einen von -1. Die Dauer des Zustands beträgt 50 % der Taktzeit, bei den restlichen 50 % wird der Pegel auf Null gezogen.

# Codierung




*Codierung nach dem AMI-Verfahren*

Bei diesem Verfahren dürfen nie zwei Signale mit gleicher Polarität aufeinander folgen. Das hat den Vorteil, dass ein gleichspannungsfreies Digitalsignal entsteht. Die AMI-Codierung wird u.a. bei DS-1 eingesetzt und in modifizierter Form bei ISDN an der S0-Schnittstelle angewandt.
Bei der modifizierten Form der *bipolaren* AMI-Codierung wird der binäre Zustand 1 durch den Signalpegel 0 Volt repräsentiert, dem binären Zustand 0 wird entweder eine negative oder eine positive Signalpolarität zugeordnet, wobei sich die Polarität zwischen zwei aufeinander folgenden 0-Signalen ändert. Der ersten 0 wird immer ein negativer Pegel zugeordnet. Zwei aufeinander folgende binäre Nullen werden durch ein alternierendes Signal aus positivem und negativem Pegel oder umgekehrt dargestellt. Dieser *Code* ermöglicht es, beliebige Bitfolgen zu übertragen, ohne dass Gleichstromanteile entstehen.

**ANSI-Code**
*ANSI code*

Der ANSI-*Code* ist ein *8-Bit-Code*, wodurch insgesamt 256 Zeichen darstellbar sind.
Der ANSI-Code umfasst neben dem deutschen Schriftsatz mit Groß- und Kleinbuchstaben, Buchstaben aus anderen Schriftsätzen, Ziffern und Sonderzeichen sowie spezielle Steuerzeichen für die textliche Bearbeitung, für Drucker und das Aufrufen von Grafiken. Er

wurde von der ANSI spezifiziert und unterstützt die Gestaltung von Benutzeroberflächen.

**Artefakt**
*artefact*

Bei der *Codierung* und Decodierung von Grafik-, Audio- und Videosignalen treten ebenso wie bei der Kompression und Dekompression Fehler auf, die als Artefakte bezeichnet werden. Diese Artefakte machen sich in Sprach-, Audio- und Video-Übertragungen als Störungen oder Geräusche bemerkbar. Typische Artefakte sind die kleinen Quadrate bei der MPEG-Kompression.

**B3ZS, bipolar with three zero substitution**
*B3ZS-Codierung*

Die B3ZS-*Codierung* ist eine *Kanalcodierung* mit *bipolaren* Violations und *AMI*-konformen Bits. Die Violation-Bits werden dann zwangsweise eingefügt, wenn der Datenstrom mehr als drei aufeinander folgende Nullen aufweist. Damit wird sichergestellt, dass der Datenstrom hinreichend viele Pegeländerungen für die Synchronisation aufweist.
Die B3ZS-Codierung entspricht im Wesentlichen der *HDB3-Codierung* und wird in der nordamerikanischen T-Leitungen (T3) eingesetzt.

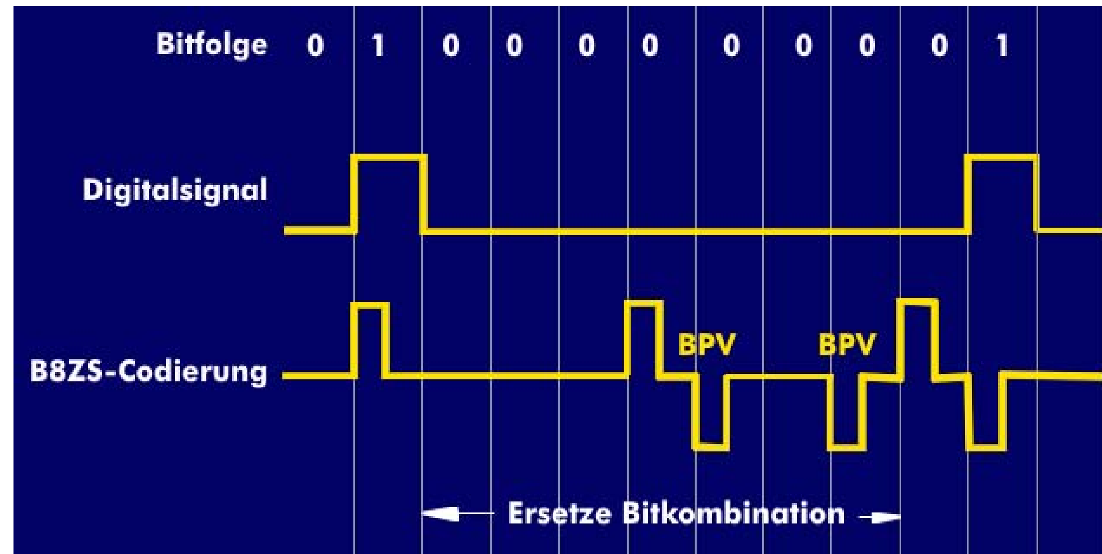
**B6ZS, bipolar with six zero substitution**
*B6ZS-Codierung*

Die B6ZS-*Codierung* ist eine *Kanalcodierung* mit *Bipolar Violation* (BPV) und *AMI*-konformen Bits. Die *bipolaren* Violations-Bits werden dann zwangsweise eingefügt, wenn der Datenstrom mehr als sechs aufeinander folgende Nullen aufweist. Damit wird sichergestellt, dass der Datenstrom hinreichend viele Pegeländerungen für die Synchronisation aufweist.
Die B6ZS-Codierung entspricht im Wesentlichen der *HDB3-Codierung* und wird in der nordamerikanischen T-Leitungen (T2) eingesetzt.

**B8ZS, binary 8 zero substitution**

*Binary* 8 Zero Substitution (B8ZS) ist eine *Kanalcodierung* mit *Bipolar* Violation (*BPV*) und *AMI*-konformen Bits. Bei der B8ZS-*Codierung* werden digitale Einsen alternativ durch einen

*B8ZS-Codierung*




*B8ZS-Codierung*

positiven und negativen Pegel repräsentiert. Der ersten 1 wird beispielsweise der positive Pegel +1 zugeordnet, dann hat die ihr folgende 1 den negativen Pegel -1. Die digitale 0 entspricht dem 0-Pegel.

Bei dieser Kanalcodierung werden acht aufeinander folgende Nullen durch zwei dicht aufeinander folgende bipolare Pulse ersetzt, dem so genannten Bipolar Violation (BPV). Darüber hinaus muss mindestens eins von acht aufeinander folgende Bits auf 1 gesetzt sein. Empfangsseitig werden die beiden unipolaren Pulse wieder in acht Nullen zurückgewandelt.

Das B8ZS-Verfahren entspricht im Wesentlichen der *HDB3-Codierung* und benutzt wie dieses das Violation. Es hat den Vorteil, dass auch bei einer kontinuierlichen Bitfolge aus Nullen die Synchronisation sichergestellt ist. Es wird in E1 und der DS-Übertragungsschnittstelle DS-1 eingesetzt.

**Barker-Code**
*barker code*

Der Barker-*Code*, der ursprünglich für Radaranwendungen entwickelt wurde, ist ein *Codierschema* für die Spreizbandtechnik (DSSS) in WLANs nach IEEE 802.11b und 802.11g.

# Codierung

| Barker-Code | Digitalsequenz |
|---|---|
| R2 | 10 oder 11 |
| R3 | 110 |
| R4 | 1011 oder 1001 |
| R5 | 11101 |
| R7 | 1110010 |
| R11 | 11100010010 |
| R13 | 1111100110101 |

*Barker-Code*

Bei diesem Codierschema ersetzt eine Bitsequenz von 13 Bits oder weniger ein einzelnes Informationsbit. Im Falle der WLANs nach 802.11 wird mit einer Bitsequenz von 11 Bits gearbeitet, wobei die normale Bitsequenz der digitalen 0 entspricht und die digitale 1 durch die inverse Bitsequenz dargestellt wird. Die sichtbarste Eigenschaft des Barker-Codes sind die unbedeutenden Spitzen der Autokorrelation, die immer aus +1, 0 und -1 bestehen.

**Base64, Base64 code**
*Base64-Code*

Base64 ist ein abgespeckter ASCII-Zeichensatz, der aus 64 Buchstaben, Ziffern und Sonderzeichen besteht. Er wird für die *Codierung* von Binärdaten zur Übertragung von E-Mails benutzt. Der gesamte Zeichenvorrat umfasst die 26 Großbuchstaben von „A“ bis „Z“, die 26 Kleinbuchstaben „a“ bis „z“, die zehn Ziffern von „0“ bis „9“ und die zwei Sonderzeichen „+“ und „/“.

**BCD-Code**
*BCD, binary coded decimal*

Der BCD-*Code* arbeitet mit 4 Bit pro Ziffer. Das System ist zwar übersichtlicher als das Dualsystem, hat allerdings den Nachteil der langen Ziffernkolonnen. So wird beispielsweise die 4-stellige Dezimalzahl 1234 im BCD-Code zu 0001 0010 0011 0100 und im Dualsystem zu 10011010010.
Jedes *Dezimalsystem* kann im Dualsystem mit den Ziffern des Dezimalsystems dargestellt werden. Der erweiterter BCD-Code ist der *EBCDIC-Code*.

# Codierung

**BCD-Interchange-Code**
*BCD interchange code*

Beim *BCD*-Interchange-*Code* handelt es sich um einen einfach belegten *6-Bit-Code*. Die Darstellungsmenge umfasst 64 Bitkombinationen für 57 Buchstaben, Ziffern und Sonderzeichen. Dieser Code wird manchmal auch als 6-Bit-Code für Asynchron-Übertragungen verwendet.

**BCH, Bose-Chaudhuri-Hocquenghem Code**
*BCH-Code*

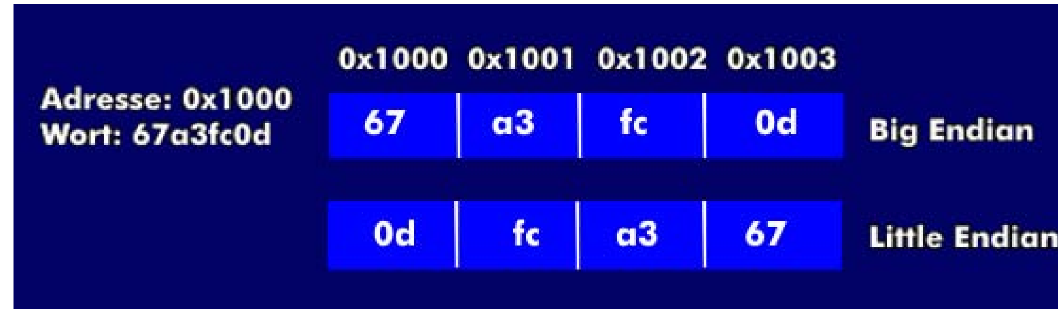
Der BCH-*Code* ist ein Code für die Fehlererkennung und Fehlerkorrektur, der mit einer *zyklischen Blockprüfung* (CRC) arbeitet. Es handelt sich um einen Multi-Level-Code mit variabler Länge mit dem bis zu 25 % der gesamten Bits überprüft und korrigiert werden können. Da der BCH-Code nur zwei Fehler korrigieren kann, werden die Fehler relativ schnell über zwei Gleichungen mit zwei Unbekannten für die Position dekodiert.
Der BCH-Code kann nicht nur für Binärsignale eingesetzt werden, sondern auch mehrpegelige Signale mit Phasenumtastung.
Eingesetzt wird der BCH-Code in der Telekommunikation, so beispielsweise in den Kontrollkanälen von zellularem *TDMA* in den USA und im POCSAG-Code.

**BER, basic encoding rules**

Basic *Encoding* Rules (BER) sind Grundregeln für die *Codierung* von Daten, die in ASN.1 beschrieben werden. Mit diesen Regeln können Datenelemente codiert werden, die zur Spezifizierung von ASN.1-Elementen als Byte-String dienen. Der String umfasst Datenfelder für den Typ, die Länge und den Wert. Das Typ-Feld zeigt die Objekt-Klasse an, das Längenfeld die Anzahl der Bytes, mit denen der Wert codiert wurde, und das Datenfeld für den Wert zeigt die Information, die mit ASN.1 in Verbindung stehen.
Die Typklasse wird von den beiden hochwertigsten Bits, Bit 8 und Bit 7, bestimmt. Es gibt folgende Typklassen: Universell (00), Anwendung (01), Kontext (10) und Privat (11).

# Codierung

**Big-Endian-Format**
*big endian*




*Big- und Little-Endian-Formate*

Bei Lade- und Speichervorgängen in Registern und Speichern werden die Bytes in unterschiedlicher Reihenfolge eingelesen. Man unterscheidet dabei zwischen dem *Little-Endian-Format* und dem Big-Endian-Format.
Das Big-Endian-Format ist ein Datenformat für die Übertragung oder Speicherung binärer Daten, in denen das höherwertige Byte oder Bit (Most Significant Byte, MSB) an erster Stelle kommt. Ein Big-Endian-System speichert höherwertige Bytes auf den niedrigsten Adressen. Mainframe-Prozessoren, einige RISC-Architekturen und Minicomputer verwenden das Big-Endian-System. In TCP/IP-Netzwerken erfolgt die Byte-Anordnung ebenfalls in Big Endian. Gegensatz: *Little Endian*. Nach dem Big-Endian-Verfahren arbeiten u.a. der MC68000 von Motorola, SPARC CPUs, IBM PowerPC.

**Binär**
*binary*

Binär oder zweiwertig bedeutet, dass ein System jeweils einen von zwei möglichen Zuständen annehmen kann, z.B. ja/nein, Strom/kein Strom, null/eins, high/low.
Die Einheit für die Zweiwertigkeit ist das Binary Digit (Bit), das die Basis für das Binär- und Dualsystem bildet.

**Binärcode**
*BC, binary code*

Binärcode ist ein *Code*, bei dem jedes *Codewort* aus Binärzeichen besteht. Da häufig Dezimalziffern *binär* codiert werden, wird die Bezeichnung *BCD-Code* verwendet.

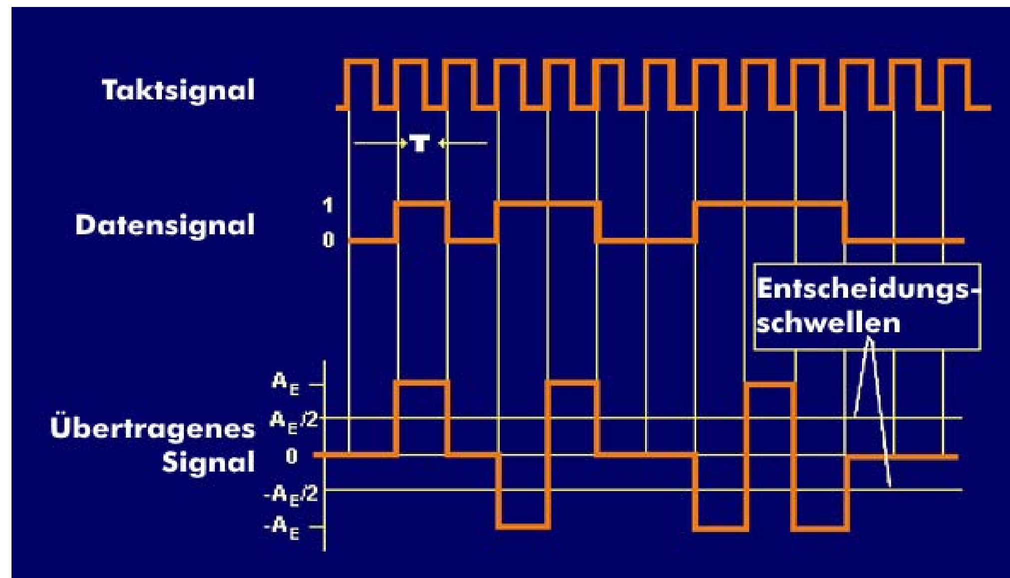
# Codierung

| Dezimal | Binär | Dezimal | Binär |
|---|---|---|---|
| 0 | 0000 | 8 | 1000 |
| 1 | 0001 | 9 | 1001 |
| 2 | 0010 | 10 | 1010 |
| 3 | 0011 | 11 | 1011 |
| 4 | 0100 | 12 | 1100 |
| 5 | 0101 | 13 | 1101 |
| 6 | 0110 | 14 | 1110 |
| 7 | 0111 | 15 | 1111 |

*Binärsystem*

Der Binärcode oder das Binärsystem bestimmt die Zuordnung der Wertigkeit der Dualzahlen zu den vierstelligen binären Codewörtern.
Der Binärcode wird auch als Zuordnungscode benutzt, so beispielsweise beim ASCII-Zeichensatz. In diesem Zeichensatz ist jeder Buchstabe, jede Ziffer, jedes Zeichen und Steuerzeichen eindeutig durch einen Binärcode gekennzeichnet.
Beispiele: Dezimalzahl 5 entspricht im BCD-Code einer 0101, Dezimalzahl 18 entspricht im BCD-Code 0001 0010.

**Bipolarverfahren**
*bipolar method*




*Bipolarverfahren*

Das Bipolarverfahren ist ein im Basisband benutztes *Übertragungsverfahren*, das dem zu übertragenden Binärwert »0« die Sendeamplitude »0« zugeordnet und der

logischen »1« alternierend die Sendeamplitude »+A« und »-A«.
Zur Rückgewinnung der Information im Empfänger müssen bei einem Empfangssignal mit den Amplituden 0 bzw. ± A die Entscheidungsschwellen auf dem Wert ± A/2 geregelt werden. Das Bipolarverfahren beseitigt die Nachteile des Einfachstrom- und des *Doppelstromverfahrens* bei der Übertragung langer »0«- bzw. »1«-Folgen, weil es keinen Gleichstromanteil enthält; es ist allerdings nicht selbsttaktend. Das Verfahren wurde ursprünglich für die *Codierung* im Ethernet entwickelt und entspricht der *AMI-Codierung*.

**Biquinärcode**
*biquinary code*

Den Biquinärcode, den es als 2-aus-5-*Code* und als 2-aus-7-Code gibt, wird für die Verschlüsselung von Dezimalziffern verwendet. Bei beiden Codes sind immer zwei Bitstellen mit „1" besetzt, die restlichen mit „0". Beim 2-aus-5-Code teilen sich die fünf Bits in zwei Gruppen mit 2 und 3 Stellen auf, wobei zwei Bits Prüfzwecken dienen. Jedes Bit der Fünfergruppe repräsentiert einen bestimmten Dezimalwert, der von links nach rechts 7, 4, 2, 1 und 0 beträgt. So wird beispielsweise die Dezimalziffer 6 durch ein „1" auf der zweiten und dritten Bitstelle gebildet, alle anderen drei Bits sind „0". Beim 2-aus-7-Code ist der Stellenwert der Bitstellen von links nach rechts 0, 5, 4, 3, 2, 1 und 0. Bei diesem Code wird beispielsweise die Dezimalziffer „0" durch eine „1" auf der ersten und der letzten Bitstelle gebildet, alle anderen fünf Bitstellen sind „0"; eine dezimale „6" würde durch eine „1" auf der dritten und fünften Stelle gebildet.

| Dezimal | Biquinär | Dezimal | Biquinär |
|---|---|---|---|
| 0 | 1000001 | 5 | 0100001 |
| 1 | 1000010 | 6 | 0100010 |
| 2 | 1000100 | 7 | 0100100 |
| 3 | 1001000 | 8 | 0101000 |
| 4 | 1010000 | 9 | 0110000 |

*Dezimalwerte im Biquinärcode*

ITWissen
Das große Online-Lexikon für Informationstechnologie

# Codierung

**BMP, basic multilingual plane**

Der Unicode verwendet in den ursprünglichen Teilmengen 16 Bit breite Zeichen für mathematische Sonderzeichen, häufig benutzte Piktogramme und Symbole. Diese Teilmenge wird als Basic Multilingual Plane (BMP) bezeichnet. Die BMP-Zeichen verwenden vier Hexadezimalzeichen.

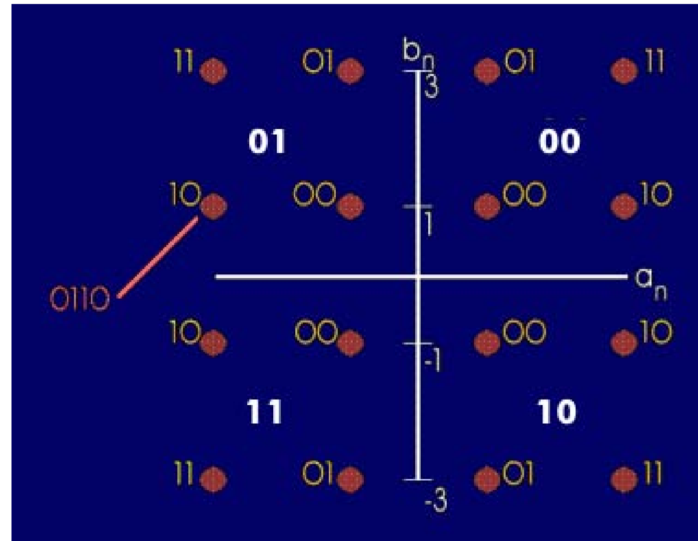
**BPV, bipolar violation**

Bei verschiedenen *Kanalcodierungen* werden mehrere aufeinanderfolgende Nullen durch eine Bitkombination unterbrochen. Man spricht in diesem Zusammenhang von Codeverletzungen oder *Code*-Violation. So beispielsweise bei *B3ZS*, *B6ZS* und *B8ZS*. Der Grund dafür liegt in der Generierung des Synchronsignals.
Diese zwangsweise eingefügten Bitkombinationen nennt man *Bipolar* Violation (BPV), sie bestehen aus zwei aufeinander folgende Pegel, einem positiven (+1) und einem negativen (-1), oder umgekehrt. Welcher Pegelzustand zuerst benutzt wird hängt von dem Pegel ab, der für die letzte 1 benutzt wurde. Wurde für die letze 1 ein positiver Pegel benutzt, dann beginnt die Bipolar Violation ebenfalls mit einem positiven Pegel. Im andern Fall mit einem negativen.

**CAP, carrierless amplitude-phase modulation**
*CAP-Codierung*

Carrierless Amplitude-Phase Modulation (CAP) ist eine kombinierte trägerlose Phasen- und Amplitudenmodulation, die z.B. zur Übertragung von ATM mit 155 Mbit/s über Kupferkabel der Kategorie 3 verwendet wird.
Bei der CAP-*Codierung* handelt es sich um eine Variante des QAM-Verfahrens, bei der die Amplitude und die Phase moduliert werden. So kann man sich ein CAP-4-System mit vier Spannungspegeln vorstellen, z.B. mit +2 V, +1 V, -1 V und -2V. Auf jeder Spannungsebene können Informationen aus zwei Bits dargestellt werden, z.B. +2 V als 00, +1 V als 01 usw. Hierdurch würden im Vergleich zu einem Zwei-Ebenen-System die Emissionsebenen für dieselbe Bitrate halbiert. Überträgt man das System auf 16 Ebenen (CAP 16), so kann auf

# Codierung




*Phasenumtastung beim CAP-16-Verfahren*

jeder Ebene eine Sequenz von vier Bits dargestellt werden, also 2exp4. Diese Technik ermöglicht im Falle einer CAP-32-Codierung eine Übertragungsgeschwindigkeit von 155 Mbit/s mit einer Übertragungsfrequenz von 31,25 MHz, weil mit jeder Schwingung 5 Bit übertragen werden. Neuere Entwicklungen wie das CAP-64 benutzen eine Übertragungsfrequenz von 64,5 MHz, was einer Datentransferrate 300 Mbit/s entspricht. CAP-16 wird als *Übertragungsverfahren* in ATM über UTP-Kabel der Kategorie 3 eingesetzt, allerdings nur bei Übertragungsgeschwindigkeiten von 51,84 Mbit/s.

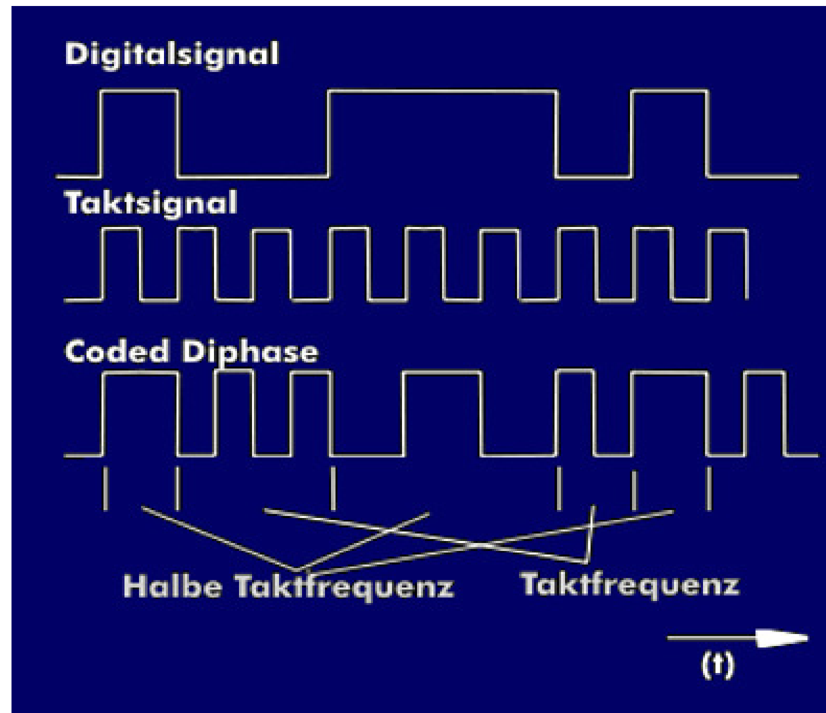
**CC, code conversion**
*Code-Konvertierung*

Unter *Code* Conversion (*CC*) versteht man die Umwandlung eines Codes, der jedes Zeichen mit einer gewissen Anzahl an Bits darstellt, in einen anderen Code, der mit der gleichen oder einer anderen Zeichendarstellung arbeitet. Es werden also die Bitgruppen eines Zeichenvorrats von einem Code in die Bitgruppen eines anderen Codes konvertiert. Beispielsweise kann ein *6-Bit-Code* beim Code Conversion in einen *8-Bit-Code* umgewandelt werden.

**CD, coded diphase**

Coded Diphase (CD) ist ein *Übertragungsverfahren* für Basisband. Dieses Verfahren arbeitet mit Phasenumtastung wobei sich bei gleich bleibendem digitalen Zustand der Pegelwert

# Codierung




*Übertragungsverfahren Coded Diphase*

entweder mit jedem Taktimpuls oder mit der halben Taktfrequenz ändert. Bei der Übertragung von Nullen wird eine Bitfolge des Taktsignals gesendet, bei *binären* Einsen wird auf die halbe Taktfrequenz umgeschaltet. Das Coded-Diphase-Verfahren ist unabhängig von dem Digitalsignal und überträgt bei kontinuierlichen Digitalpegeln immer eine Bitfolge: Bei Null die Taktfrequenz, bei Eins die halbe Taktfrequenz.

**CMI, coded mark inversion**
*CMI-Code*

Coded Mark Inversion (CMI) ist ein Verfahren für die *Codierung* binärer Signale. Dieser *Code* basiert auf einer Änderung des codierten Zustandes während der Taktzeit des Binärsignals.
Eine digitale »0« wird danach in der ersten Hälfte der Taktzeit mit der negativen Amplitude dargestellt, in der zweiten Hälfte der Taktzeit wird die »0« mit der positiven Amplitude dargestellt. Die digitale »1« wird dagegen während der gesamten Taktzeit entweder mit dem positiven Wert oder mit dem negativen Amplitudenwert dargestellt; allerdings alternierend. Dadurch wird in der Codierung eine Gleichstromfreiheit in der Übertragung erreicht. Die CMI-Codierung wird beispielsweise bei den E-Übertragungsschnittstellen für E-4 eingesetzt.

# Codierung

**Code**
*code*

Nach DIN 43000 ist ein Code eine Vorschrift für die eindeutige Zuordnung von Zeichen eines Zeichenvorrats zu denen eines anderen Zeichenvorrats. Für die Zuordnung der Zeichen des einen Zeichenvorrats zu denen eines anderen, setzt man auf solche Zeichensätze, mit denen die beste Darstellung zu erzielen ist.
Der Code mit dem kleinstmöglichen Zeichenvorrat ist der *Binärcode* mit den zwei Zeichen »0« und »1«. Für die synchrone Datenübertragung werden am häufigsten der ASCII-Zeichensatz und der *EBCDIC-Code* eingesetzt.
Codes lassen sich nach bestimmten Charakteristiken wie dem Aufbau klassifizieren und werden in allen technischen Disziplinen eingesetzt. In der Warenwirtschaft mit den Strichcodes und den 2D-Codes, im Geldverkehr mit dem BIC-Code oder der IBAN, in der Datenübertragung mit den diversen Codes für die Signalanpassung und Transferoptimierung oder in den Funktechniken mit den *Faltungscodes* und den vielen Modulationscodes.

**Code-Element**
*code element*

Ein *Code*-Element ist der kleinste Teil eines Codewortes, das zwei oder mehr verschiedene Werte annehmen kann. Ein Code-Element wird durch die Anzahl diskreter Stufen oder Zustände spezifiziert. Ein Code-Element mit 2 Zuständen heißt *binär* (Binary), mit 3 Zuständen *ternär* (Ternary), mit 4 Zuständen quarternär (Quaternary), mit 5 Zuständen quinär (Quinary), mit 6 Zuständen senär (Senary), mit 7 Zuständen septenär (Septenary), mit 8 Zuständen oktonär (Oktonary), mit 9 Zuständen novonär (Novonary), mit 10 Zuständen denär (Denary) und mit n Zuständen n-när (n-nary).

**Coderate**
*code rate*

Um eine möglichst fehlerfreie Übertragung von Daten sicher zu stellen, werden bei der *Kanalcodierung* zusätzliche Bits für die Vorwärts-Fehlerkorrektur eingefügt. Dieses Verhältnis zwischen der Anzahl an Informationsbits und Codebits ist die Coderate. Werden viele

Redundanzbits eingefügt wie beim *Faltungscode*, dann ist die Coderate gering, da die Anzahl der Codebits gegenüber den Informationsbits hoch ist. Werden beispielsweise für jeweils drei Bits ein zusätzliches Redundanzbit eingefügt, dann beträgt die Coderate 3/4.

**Codewert**
*code value*

Als Codewert bezeichnet man die Anzahl der Bits, die zur Verschlüsselung eines Daten- oder Steuerzeichens verwendet werden. Wird zur Datensicherung gegen Übertragungsfehler die Methode der Zeichenparität verwendet, so muss sie für die Berechnung der Zeichengeschwindigkeit in der Größe des Codewerts berücksichtigt werden.

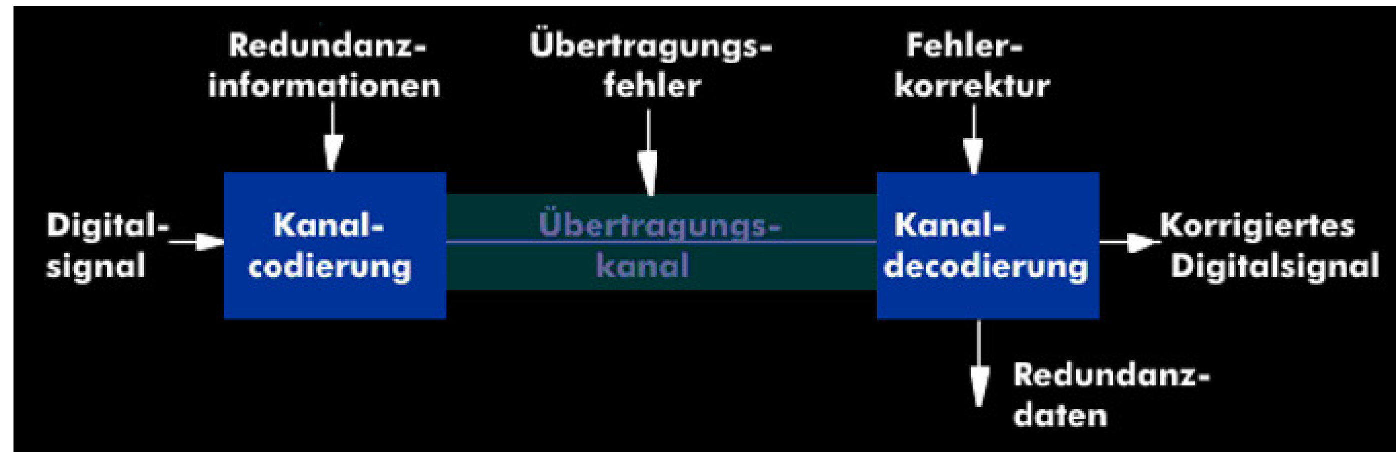
**Codewort**
*codeword*

Als Codewort bezeichnet man die Verbindung mehrerer *Code-Elemente*. Die vereinbarte Anzahl von *Code*-Elementen kann beliebig groß sein. Ein m-stelliges Codewort besteht demnach aus m Code-Elementen. Je nach Anzahl der diskreten Stufen oder Zustände des Code-Elements kann ein m-stelliges Codewort (n expm) Kombinationen bilden.
Codewörter können aus einer Zeichen- und Symbolsequenz bestehen und müssen die spezifischen Regeln des Codes einhalten. Codewörter werden u.a. in den Fehlererkennungs- und Fehlerkorrekturcodes verwendet und in der Kommunikation. Eines der bekanntesten Codewörter ist das aus dem internationalen Telegrafenalphabet bekannte SOS.
Ein Codewort kann auch ein verschlüsseltes Wort sein, mit dem sensitive Daten übertragen werden.

**Codierung**
*coding*

Das Wort Codierung wird vorwiegend in der angelsächsischen Literatur verwendet, in der deutschsprachigen Literatur spricht man von *Übertragungsverfahren*. Ziel der Codierung ist es, die Daten mit größtmöglicher Qualität zu übertragen und beim Benutzer darzustellen.
Bei der Codierung wird jedes einzelne Zeichen eines bestimmten Zeichenvorrats in das

# Codierung




*Prinzip der Kanalcodierung*

entsprechende Zeichen eines anderen Zeichenvorrats übertragen. Jedem Signalelement können dabei verschiedene Werte oder Zustände zugeordnet werden. Außer der Zustandsdefinition können den Signalen auch mehrpegelige Spannungswerte bestimmter Zeitdauer zugeordnet werden.

Bei der Übertragung werden die zu übertragenden Daten in einem Coder codiert und vom Sender übertragen. Empfangsseitig wird das empfangene Datensignal zuerst in einem *Decoder* decodiert und steht dann wieder in der ursprünglichen Form zur Verfügung. Für die Codierung existieren eine Reihe von Codierverfahren, auch Leitungscode genannt.

Man unterscheidet zwischen der *Quellencodierung* und der *Kanalcodierung*, zwischen der verlustfreien und der verlustbehafteten Codierung.

Bei der Quellencodierung wird das analoge Signal mit möglichst geringen Verfälschungen in ein Digitalsignal gewandelt. Die Kanalcodierung sorgt hingegen für die Anpassung des übertragenen Digitalsignals an die Übertragungseigenschaften des Kanals.

# Codierung

**DBP, differential biphase coding**
*DBP-Codierung*

Das Differential Biphase *Coding* (DBP) ist eine Flankencodierung. Dabei werden nur die *binären* Nullen als Halbbit-Periode codiert. Die Amplitude ändert sich bei der halben Bitperiodendauer von „0" nach „1" und fällt mit der folgenden Flanke wieder auf „0". Liegt eine binäre Eins an, dann bleibt das Codiermuster unverändert.
Damit der Empfänger das Taktsignale leichter regenerieren kann, wird das DBP-Signal jeweils zu Periodenbeginn invertiert.

**Decoder**
*DEC, decoder*

Nach DIN 44300 ist ein Decoder ein *Code*-Umsetzer und damit eine Funktionseinheit, die den Zeichen eines Zeichenvorrats eindeutig die Zeichen eines anderen zuordnet.
Schaltungstechnisch ist ein Decoder eine Einrichtung, die eine *Codierung* rückgängig macht, also decodiert.
In Satelliten-Empfangseinrichtungen handelt es sich um ein Gerät, dass mit verschlüsselte Signale decodiert und dem Benutzer den Dienstzugang ermöglicht.

**DER, distinguished encoding rule**

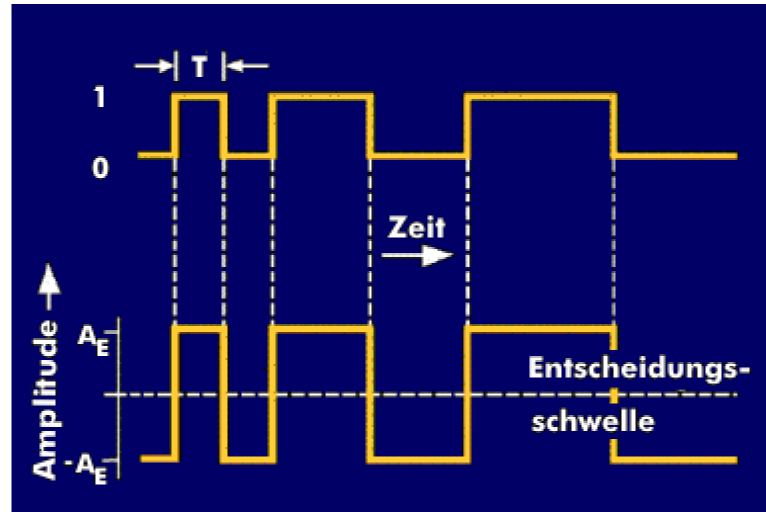
Die Distinguished *Encoding* Rules (DER) sind ein Subset der *Basic Encoding Rules* (BER), bei der es nur eine Möglichkeit gibt einen beliebigen Wert in ASN.1 als Oktett darzustellen. Die Distinguished Encoding Rules sind in der ITU-Empehlung X.690 beschrieben.

**Differenzielle Manchester-Codierung**
*differential manchester coding*

Die differenzielle *Manchester-Codierung* wird bei Token-Ring-Netzen genutzt und ist unter IEEE 802.5 spezifiziert.
Bei dieser *Codierung* wechselt der Zustand, wie bei der Manchester-Codierung, jeweils in der Bitmitte. Die Übergangsrichtung wechselt immer bei einer »Eins«. Diese Codierung ergibt sich aus einer XOR-Verknüpfung von Taktsignal und einem *NRZ-I*-codiertem Signal.

# Codierung

**Doppelstromverfahren**
*double-current technique*




*Doppelstromverfahren*

Das Doppelstromverfahren ist ein *Übertragungsverfahren* in Basisbandsystemen. Bei diesem Verfahren werden, im Gegensatz zum *Einfachstromverfahren*, den übertragenen Digitalsignalen zwei Spannungspegel zugeordnet. Die logische »1« ist durch eine positive Spannung gekennzeichnet, die logische »0« durch eine negative Spannung. Die Entscheidungsschwelle ist beim Doppelstromverfahren der Nulldurchgang.

Nachteilig bei diesem Verfahren ist, dass lange 0- oder 1-Folgen als Gleichstromanteil mit übertragen werden müssen, was in der Praxis zu Schwierigkeiten führt.

**DSS, distributed sample scrambler**
*DSS-Codierung*

Bei dem DSS-Verfahren handelt es sich um ein Verschlüsselungsverfahren für den Nutzdatenbereich von ATM-Zellen. Da bei der direkten Zellenübertragung (Cell Based Physical Layer) keinerlei Anpassung des Zellenformats auf die Übertragungsrahmen der jeweiligen Übertragungsleitung durchgeführt wird, also die Zellen Bit für Bit direkt in die elektrischen bzw. optischen Signale für das jeweilige Übertragungsmedium umgewandelt werden, wird das Informationsfeld mit den Nutzdaten verschlüsselt übertragen. Das dafür verwendete *Codierverfahren* Distributed Sample *Scrambler* (DSS) arbeitet mit einem Generatorpolynom Xexp31 + Xexp28 +1.

ITWissen
Das große Online-Lexikon für Informationstechnologie

# Codierung

**Duobinär**
*duobinary*




*Duobinäre Codierung*

Duobinär ist eine pseudobinäre *Codierung*, die mit zwei verschiedenen Hi-Leveln arbeitet, einem positiven und einem negativen. Die logische 0 wird durch einen Null-Pegel dargestellt, die logische 1 durch einen positiven oder negativen Pegel. Welcher Pegelzustand dargestellt wird hängt von der Anzahl der logischen Null-Zustände nach dem letzten 1-Zustand ab. Bei einer geraden Anzahl an Nullen seit dem letzten 1-Pegel, wird der positive 1-Pegel genutzt, bei einer ungeraden Anzahl an Nullen wird die logische 1 durch den negativen 1-Pegel dargestellt.
Duobinäre Signale benötigen weniger Bandbreite als die *NRZ-Codierung*. Außerdem können einige Fehler ohne zusätzliche Paritätsbits erkannt werden.

**EBCD, extended binary coded decimal**
*6-Bit-Code*

Extended *Binary Coded Decimal* (EBCD) ist ein Zeichencode aus sechs Daten- und einem Kontrollbit, der von Hollerith entwickelt und in Lochkarten angewendet wurde. Daher auch die Bezeichnung 6-Bit-*Code*. EBCD unterscheidet sich gegenüber Binary Coded Decimal (BCD) durch die Nutzung zusätzlicher Zonen und nicht-standardisierter Stanzkombinationen.

# Codierung

**EBCDIC, extended binary coded decimal interchange code**
***EBCDIC-Code***

Der *Extended Binary Coded Decimal* Interchange *Code* (EBCDIC) ist ein erweiterter *8-Bit-Code*, der von IBM entwickelt wurde und in IBM-Mainframes als interner Verarbeitungscode angewendet wird. Er kann ohne Codeumsetzung z.B. mit dem genormten ASCII-Zeichensatz für die Datenübertragung eingesetzt werden. Der EBCDIC-Code besteht aus Buchstaben, Ziffern, Steuerzeichen und grafischen Symbolen.

| Zeichen | Zonen-Bits | Ziffern-Bits |
|---|---|---|
| | Z Z Z Z<br>8 4 2 1 | 8 4 2 1 |
| A<br>B<br>C<br>D | 1 1 0 0<br>1 1 0 0<br>1 1 0 0<br>1 1 0 0 | 0 0 0 1<br>0 0 1 0<br>0 0 1 1<br>0 1 0 0 |

*Aufbau des EBCDIC-Codes mit Zonen- und Ziffernteil*

Bei EBCDIC sind die Bytes in einen Zonen- und einen Ziffernteil unterteilt, wobei jedes Byte in zwei *Quadbits* oder Nibbles unterteilt ist. Ein Quadbit steht für den Zonenteil, das zweite für den Ziffernteil. Die darstellbaren Zeichen sind in Gruppen eingeteilt, die Gruppencodierung erfolgt im Zonenteil. Innerhalb einer Gruppe sind die Zeichen durchnummeriert und werden im Ziffernteil spezifiziert. Da bei EBCDIC nicht allen 256 Kombinationsmöglichkeiten Buchstaben, Ziffern oder Zeichen zugeordnet sind, können die nicht belegten Kombinationen für anwenderspezifische Implementierungen genutzt werden. Von EBCDIC existieren verschiedene Varianten, die untereinander inkompatibel sind. Die amerikanische Variante benutzt weitgehend die gleichen Zeichen wie der ASCII-Code. Einige Zeichen sind aber in dem jeweils anderen Code nicht enthalten.
Eine Umwandlung vom EBCDIC-Code in ASCII ist nur über Codierungstabellen möglich.

**EFM, eight to fourteen modulation**

Die Eight to Fourteen Modulation (EFM) ist eine *Codierung*, die bei Compact Discs (*CD*) und DVDs verwendet wird und für ein einwandfreies Lesen des Speichermediums zuständig ist.

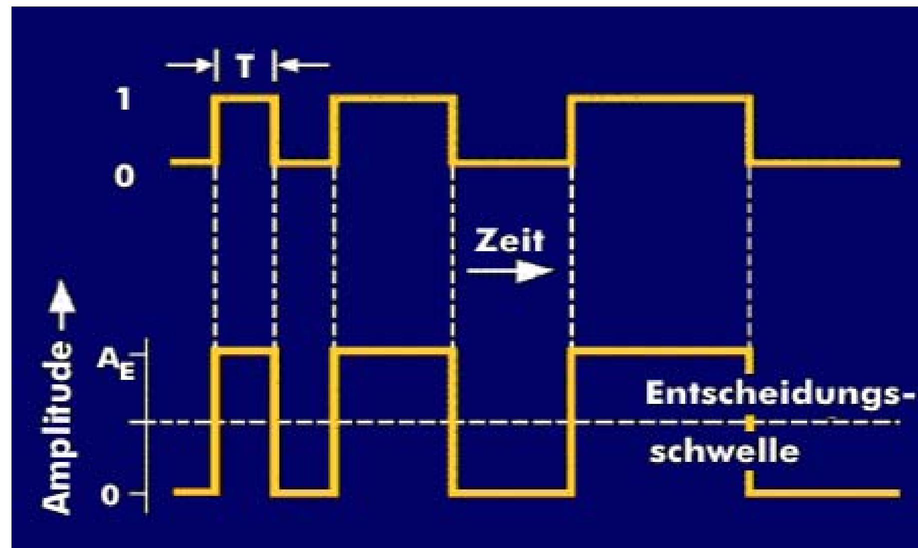
# Codierung

| 8-Bit-Daten | 14-Bit-Bitmuster (EFM) |
|---|---|
| 00000000 | 01001000100000 |
| 00000001 | 10000100000000 |
| 00000010 | 10010000100000 |
| 00000011 | 10001000100000 |
| 00000100 | 01000100000000 |
| 00000101 | 00000100010000 |
| 00000110 | 00010000100000 |
| 00000111 | 00100100000000 |

*Beispiel einer EFM-Codierung*

Beim EFM-Verfahren werden jeweils 8 Bit in ein Bitmuster mit 14 Bit umgesetzt, wobei zwischen den zwei Einsen mindestens zwei und höchstens elf Nullen stehen dürfen.
Das EFM-Verfahren ist in CD-Laufwerken implementiert. Eine Variante dieses Verfahrens ist EFM-Plus, das in DVDs angewendet wird und wegen einer erhöhten Datensicherheit mit 16 physikalischen Zuständen pro Byte arbeitet. EFM-Plus arbeitet mit 3 Kontrollbits pro Byte.

**Einfachstromverfahren**
*neutral current method*




*Prinzip des Einfachstromverfahrens*

Beim Einfachstromverfahren werden nur Signale mit positiven oder negativen Spannungen übertragen.
Bei diesem *Übertragungsverfahren* für Basisbandsysteme pegelt sich die Entscheidungsschwelle, ob es sich bei dem zu übertragenden Digitalsignal um eine 1 oder eine 0 handelt, auf dem halben Pegelwert (zwischen 0 und 1) ein. Nachteilig ist bei diesem

# Codierung

Verfahren, dass Störungen als logische Werte interpretiert werden können.

**Exzess-3-Code**
*excess 3 code*

| Dezimal-zahl | Exzess-3-Code | Dezimal-zahl | Exzess-3-Code |
|---|---|---|---|
| 0 | 0011 | 8 | 1011 |
| 1 | 0100 | 9 | 1100 |
| 2 | 0101 | 10 | 1101 |
| 3 | 0110 | 11 | 1110 |
| 4 | 0111 | 12 | 1111 |
| 5 | 1000 | 13 | 0000 |
| 6 | 1001 | 14 | 0001 |
| 7 | 1010 | 15 | 0010 |

*Exzess-3-Code*

Der Exzess-3-*Code* dient wie das Binärsystem oder der *Aiken-Code* der Darstellung von Dezimalzahlen in binärer Form. Der Unterschied zum Binärsystem besteht darin, dass der Exzess-3-Code mit einer Vorgabe von „3", bzw. der Binärzahl 0011, arbeitet.
Die Dualzahlen des Exzess-3-Codes zeigen bei allen Dualwerten diesen Versatz von 0011. So hat die Dezimalzahl „0" den Dualwert 0011, die Dezimalzahl „1" den Dualwert 0100 usw.
Der Exzess-3-Code bietet Vorteile bei der Addition von Dualwerten, weil der Zehnerübertrag simuliert wird.

**Gray-Code**
*Gray code*

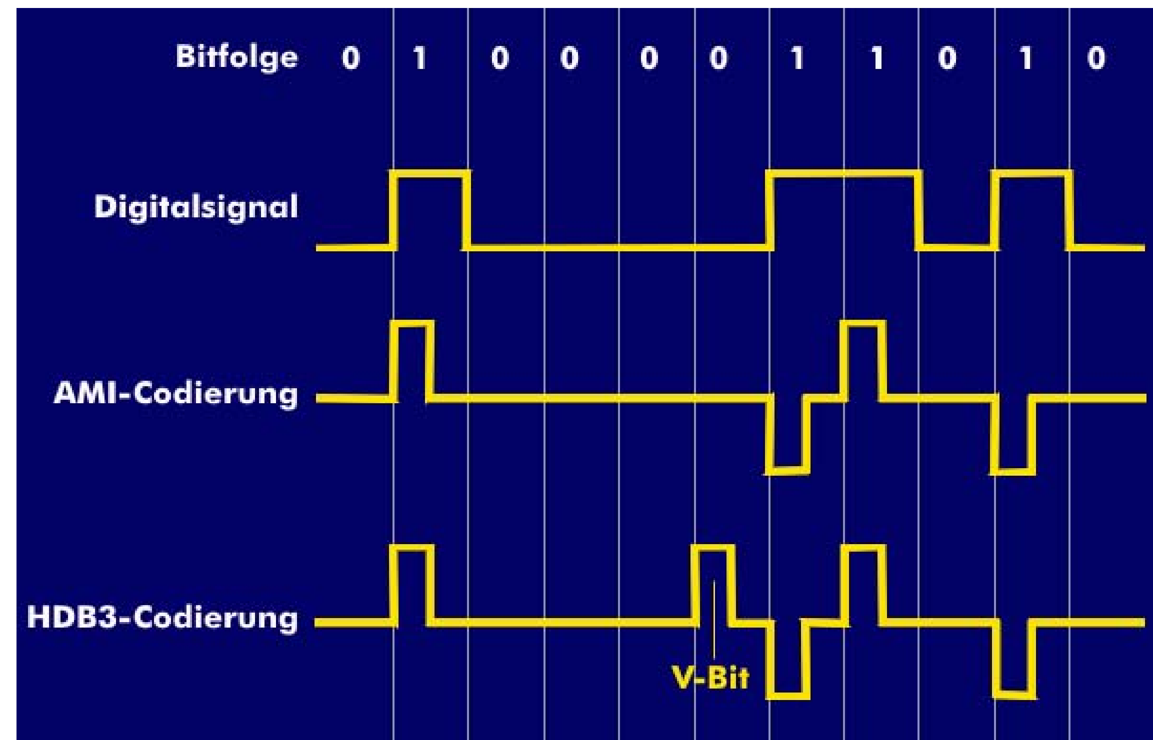
Der Gray-*Code* ist nach dem amerikanischen Physiker Frank Gray benannt. Es ist ein einfacher Code, bei dem sich zwei aufeinander duale *Codewörter* nur in einem einzigen Bit unterscheiden dürfen. Der Gray-Code hat somit zwischen zwei aufeinander folgenden Binärwörter eine konstante Hamming-Distanz von 1. Die Codewörter des Gray-Codes können nur zwei Bit umfassen, womit vier Codewörter möglich sind, aber ebenso 3, 4, 5 oder 6, was 64 Codewörter bedeutet.
Übertragungsfehler sind daran zu erkennen, dass eine andere als die theoretisch vorgegebene

# Codierung

Reihenfolge der um 1 Bit versetzten Codewörter am Ende der Übertragungsstrecke ausgelesen wird. Bei einem 3-Bit-Gray-Code wäre die theoretische Reihenfolge der Codewörter 000, 001, 010, 011, 100, 101, 110, 111. Fehlerbehaftet könnte der Gray-Code so aussehen: 000, 001, 011, 010, 011, 110, 100, 101, 110, 111. Die beiden Binärwerte 011 und 110 würden als Fehler auftreten.

**HDB, high density bipolar**
*HDB-Codierung*

Die HDB-*Codierung* (High Density *Bipolar*) ist eine Weiterentwicklung der pseudoternären *AMI-Codierung* und geht aus dem einfachen *Bipolarverfahren* hervor. Der Pseudoternärcode kennt drei Signalpegel: positiver Pegel, Null-Pegel und negativer Pegel (+1, 0, -1). Die Codierung entspricht im Wesentlichen der AMI-Codierung. Bei der HDB-Codierung



*Codierung nach dem HDB3-Verfahren*

# Codierung

wird eine *binäre* „0“ durch den Null-Pegel repräsentiert. Eine binäre 1 wird durch einen alternierenden Puls dargestellt, wobei die erste 1 hat einen negativen Pegel (-1) hat, die folgende 1 wird durch +1 dargestellt. Die Dauer der Zustände beträgt 50 % der Taktzeit, in den anderen 50 % wird der Zustand wieder auf 0 gesetzt. Um die Übertragung langer Nullfolgen zu vermeiden, was die Taktrückgewinnung erschwert, wird bei der Übertragung von mehr als n aufeinander folgenden Nullen eine besondere Bitkombination, das so genannte Violation, eingefügt.
Beim HDB2-Verfahren dürfen nicht mehr als zwei Nullen aufeinander folgen (n=2). Danach wird das Violation-Bit eingefügt. Beim HDB-Verfahren hat der Pegel dieses Bits, die Polarität, wie die letzte 1. Wurde die letzte 1 mit +1 dargestellt, dann ist der Violation-Pegel ebenfalls +1.
Beim HDB3-Verfahren mit n = 3, wird nach drei Nullen das Violation-Bit hinzugefügt, entweder als +1 oder als -1. Werden zwischen zwei Violation-Zuständen eine geradzahlige Anzahl von binären Einsen übertragen, dann wird die erste 0 mit einem Puls dargestellt. Dieser Zustand entspricht einem A-Zustand, einer AMI-konformen Regelverletzung.
Der HDB-*Code* wird bei der Übertragung von pulscodemodulierten Signalen benutzt; u.a. wird er für die Übertragungsschnittstellen E1 und E3 eingesetzt.

**Huffman-Codierung**
*Huffman encoding*

Die Huffman-*Codierung* ist ein verlustfreies Codierverfahren für die Text- und Bildkompression. Das Verfahren basiert auf der Redundanz, der Zeichenhäufigkeit und der statistischen Verteilung der Daten einer Information. So treten beispielsweise in einem Text bestimmte Buchstaben häufiger auf als andere, so der Buchstabe „e“, der häufiger vorkommt als andere. In einer Grafik sind hingegen die Farben statistisch verteilt: Grün kann beispielsweise häufiger vorkommen als Gelb.

# Codierung

Bei der Huffman-Codierung, die nach dessen Erfinder David Huffman (1925-1999) benannt ist, ersetzt man den am häufigsten vorkommenden Datensatz durch einen kurzen Signalcode, seltener vorkommende Datensätze werden hingegen durch längere Signalcodes ersetzt. Die am häufigsten auftretenden Zeichen werden mit nur 3 Bit (Short Hand) übertragen, das bedeutet, dass man bei einem Zeichen 5 Bit einspart. Manche, selten auftretende Zeichen werden sogar mit mehr als acht Bit übertragen.
Die Länge des *Codes* ist also variabel und wird in der Huffman-Codierung anhand der Häufigkeit festgelegt. Dabei wird die Häufigkeit der einzelnen Buchstaben oder Farbpixel ermittelt, Buchstaben/Farbpixel mit geringerer Häufigkeit werden in Gruppen zusammengefasst und aus denen werden in einer Baumstruktur wiederum neue Gruppen gebildet. Je nach Zusammensetzung der Signale können so bis zu 50 Prozent und mehr der Übertragungszeit eingespart werden.
Die Huffman-Codierung wird im H.320 und anderen Videocodecs, in der Gruppe-3 Fax, bei MNP 5 und in JPEG verwendet.

**Little-Endian-Format**
*little endian*

Bei Lade- und Speichervorgängen in Registern und Speichern gibt es zwei Anordnungssysteme in dem die Bytes eingelesen werden: Das Little-Endian-Format und das *Big-Endian-Format*. Im Gegensatz zu Big Endian ist Little Endian ein Format für die Übertragung oder

| Adresse: 0x1000<br>Wort: 67a3fc0d | 0x1000 | 0x1001 | 0x1002 | 0x1003 | |
|---|---|---|---|---|---|
| | 67 | a3 | fc | 0d | Big Endian |
| | 0d | fc | a3 | 67 | Little Endian |

*Big- und Little-Endian-Formate*

# Codierung

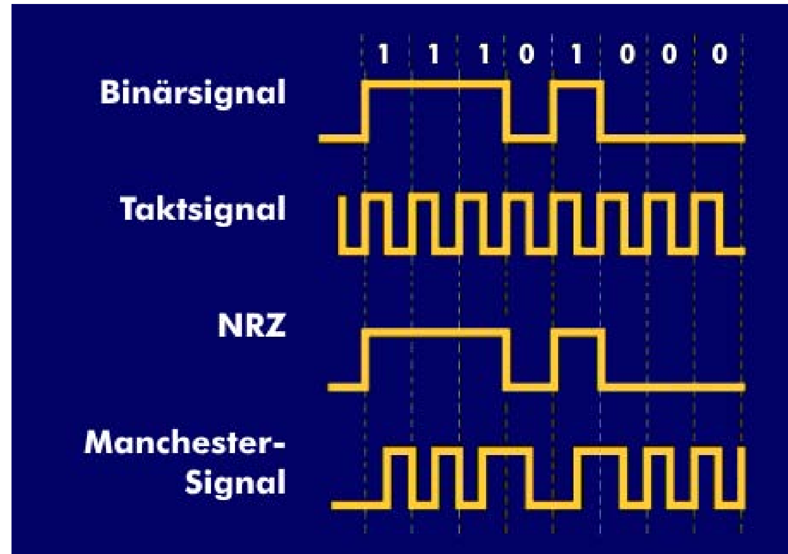
Speicherung binärer Daten, in denen das Least Significant Byte (LSB) an erster Stelle kommt und auf der niedrigsten Speicheradresse gespeichert wird. So wird das Byte E3 von der hexadezimalen Zahl 6FE3 im Little-Endian-Format auf dem Speicherplatz mit der niedrigsten Adresse gespeichert.
Nach diesem Prinzip arbeiten u. a. Intel, die VAX von *DEC*, DEC-Alpha und MIPS-Chips.

**LPC, linear predictive coding**
*LPC-Codierung*

Linear Predictive *Coding* (LPC) ist ein Transformationsverfahren, das in der Sprachkompression von Mobilfunksystemen eingesetzt wird. Das Verfahren arbeitet mit Filtertechniken und filtert aus einem Signalgemisch sprachrelevante Komponenten wie die Sprachgrundfrequenz, Lautstärke und Stimmhaftigkeit heraus.
Dazu wird das digitale Sprachsignal in ein Erregersignal und in Koeffizienten eines zeitvarianten Filters transformiert. Dies geschieht durch regelmäßige Abtastung (160 Abtastwerte pro Sekunde) des zu übertragenden Sprachsignals, aus dem dann Filterkoeffizienten gebildet und zum *Decoder* übertragen werden. Bei der Aufteilung in kurzzeitige Segmente geht man davon aus, dass die Änderung zwischen zwei Segmenten marginal ist. Durch die lineare Vorhersage brauchen daher nur die Änderungen zwischen zwei aufeinander folgende Kurzzeitsegmente übertragen zu werden, wodurch die Dateigröße wesentlich reduziert werden kann.
Mit der LPC-Codierung können bei Abtastraten von 8 kHz und 16 kHz Datenraten von 6 kbit/s bis 24 kbit/s erreicht werden. Das Verfahren wird u.a. auch zur Sprachcodierung in GSM eingesetzt.
Der Unterschied zwischen der Linear Predictive Coding (LPC) und der Adaptive Predictive Coding (APC) liegt darin, dass APC weniger Vorhersagekoeffizienten benutzt und daher eine höhere Abtstrate bietet als die LPC-Codierung.

# Codierung

**Manchester-Codierung**
*Manchester encoding*



*Manchester-Codierung*

Bei der Manchester-*Codierung* werden die *binären* Informationen durch Spannungswechsel innerhalb der Bitzeit dargestellt. Dadurch können Sender und Empfänger wesentlich leichter synchronisiert werden, denn der Übergang in der Mitte der Bitzeit ergibt einen zuverlässigen Takt.
Die erste Hälfte der Bitzeit enthält die Repräsentation des zu übertragenden komplementären Bitwertes, die zweite Hälfte repräsentiert den Bitwert (spezifiziert für IEEE 802.3 Ethernet).
Die Manchester-Codierung stellt eine XOR-Verknüpfung aus Taktsignal und *NRZ*-codiertem Signal dar, bei der eine Verdoppelung der Frequenz stattfindet.

**Markov-Codierung**
*Markov encoding*

Die Markov-*Codierung* baut auf der *Huffman-Codierung* auf. Bei diesem Verfahren wird eine statistische Vorhersage über den zukünftigen Datenstrom gemacht und darauf basierend die mit 3 Bit codierten (Short Hand) Zeichen der Huffman-Codierung modifiziert. Das Verfahren ist eine Effizienzsteigerung der Huffman-Codierung.

**Middle-Endian-Format**
*middle endian*

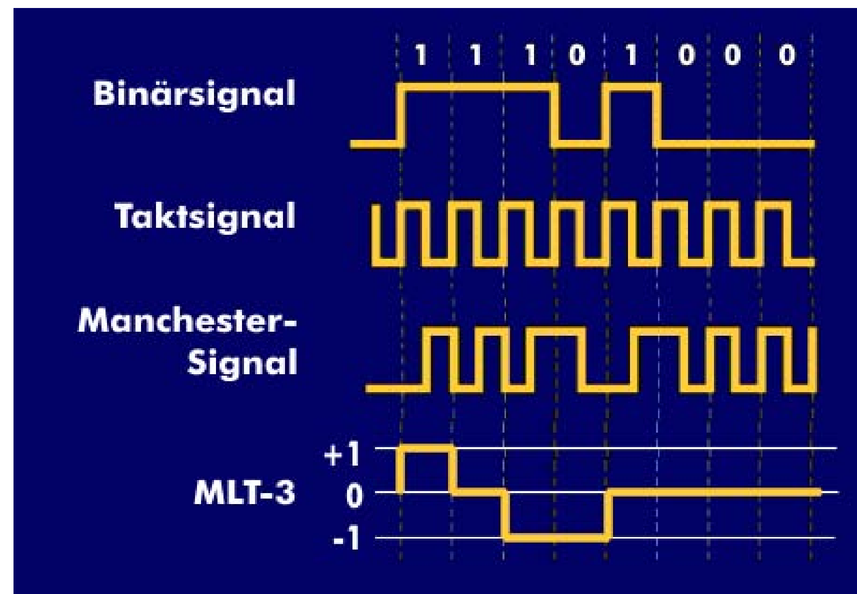
Middle Endian ist ein Datenformat für die Übertragung oder Speicherung binärer Daten. Bei *Big Endian* wird das hochwertigste Bit oder Byte auf den niedrigsten Adressen gespeichert. Bei *Little Endian* wird das geringwertigste Bit oder Byte auf niedrigen Speicheradressen

# Codierung

gespeichert und bei Middle Endian erfolgt die Speicherung irgendwo zwischen Big und Little Endian.

**MLT, multi level transmission**
*MLT-Codierung*

Die MLT-*Codierung*, Multi Level Transmission, ist ein Ternärverfahren, das für die Codierung dreiwertige Signalpegel verwendet (+V, 0V, -V), deswegen wird es auch als MLT3-Codierung bezeichnet, im Gegensatz zu den bekannten Zweipegel-Verfahren *NRZ*, *NRZ-I* und anderen. Der Vorteil dieses Verfahrens liegt in der Reduzierung der Basisfrequenz, was geringere Anforderungen an die Verkabelung zur Folge hat. Darüber hinaus werden Abstrahlungen vermindert und Bitfehlerraten minimiert. Bei den Ternärverfahren muss allerdings der Empfänger zwischen drei Pegelwerten unterscheiden.



*MLT-3-Codierung*

Das MLT-Verfahren wird in Form des MLT-3 bei FDDI über TP-Kabel benutzt. Dabei werden in einer Übertragungsschwingung 4 Bit übertragen. Das bedeutet, dass sich die eigentliche Übertragungsfrequenz auf ein Viertel des eigentlichen Wertes reduziert. Das übertragene Signal erfährt nur bei einer logischen »1« eine Pegeländerung. Ist der Pegel bei der ersten logischen »1« positiv, so wird er bei der folgenden logischen »1« zu Null und bei der darauf folgenden negativ. Da bei

# Codierung

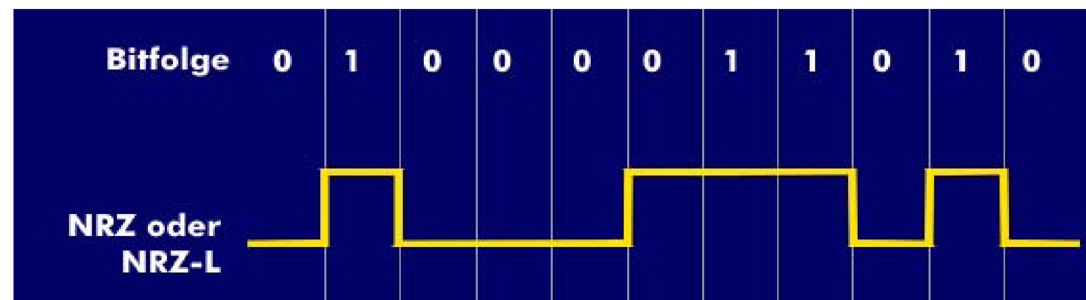
FDDI 100 Mbit/s für die Nutzdatenübertragung und 25 Mbit/s für Kontrolldaten benutzt werden, ergibt sich eine Gesamtübertragungsrate von 125 Mbit/s, bzw. eine Übertragungsfrequenz von 31,25 MHz auf dem Kabel.

**NCI, non-coded information**
*Nicht-codierte Information*

Wenn Texte oder Dokumente in einer Form ausgegeben werden, die keinen Bezug zur ursprünglichen *Codierung* zulässt, spricht man von nicht-codierten Informationen. Das können beispielsweise Präsentationen, Papierausdrucke oder Faxe sein, die keinen Rückschluss auf die Codierung zulassen.
Non-Coded Informationen (NCI) eignen sich nicht für die Archivierung. Erst durch intelligente Zeichenerkennung (ICR) oder optische Zeichenerkennung (OCR) kann ein solches Dokument erfasst und digitalisiert werden und steht dann für die Archivierung zur Verfügung.

**NRZ, non return to zero**
*NRZ-Codierung*

Non Return to Zero (NRZ) ist ein äußerst triviales *Codierverfahren* für pulsecodemodulierte Daten. Es entspricht NRZ-L. Bei NRZ und NRZ-L werden die Binärsignale direkt abgebildet. Ein 1-Bit-Signal durch einen hohen Pegel, ein 0-Bit-Signal durch einen niedrigen oder negativen Pegel dargestellt. NRZ kann mit dem *Einfachstromverfahren* und mit dem *Doppelstromverfahren* eingesetzt werden. Die Bitfolgen 10 und 01 können so durch einen einzigen Frequenzzyklus dargestellt werden. Zwei Bits werden somit



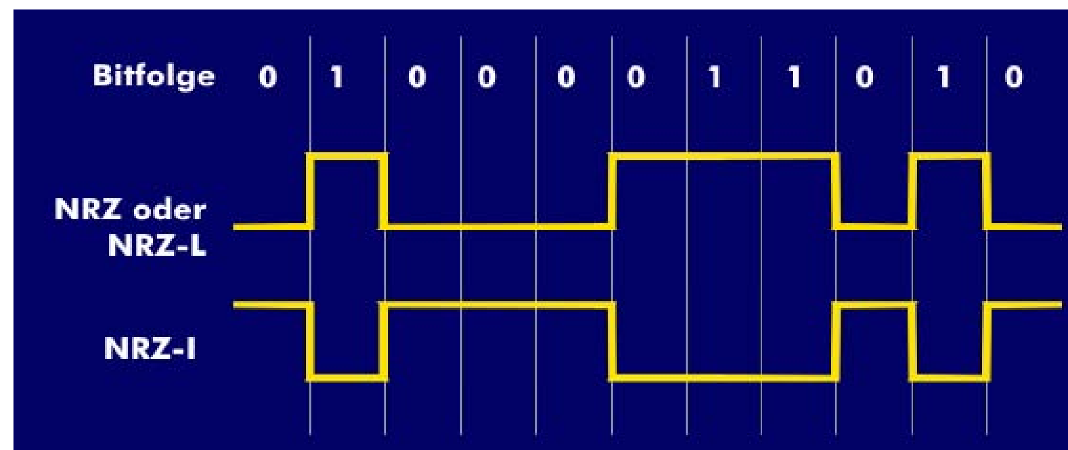

*NRZ-Codierung*

# Codierung

in ein Hertz Übertragungsfrequenz codiert. Bei aufeinander folgenden Bits von gleichem logischem Wert bleibt der Signalpegel unverändert. Das Verfahren ist nicht selbsttaktend, d.h. es besitzt keine Synchronisiereigenschaften; es erlaubt höchste Datenraten, eignet sich aber nur für kurze Übertragungsstrecken. Nachteilig sind bei der Übertragung langer 0- oder 1-Sequenzen die Schwankungen im Gleichspannungspegel sowie die Taktrückgewinnung zum Zwecke der Synchronisierung.

**NRZ-I, non return to zero inverted**
*NRZ-I-Codierung*

Bei dem zweiwertigen *Codierverfahren NRZ*-I (Non *Return to Zero Inverted*) handelt es sich um eine Invertierung von NRZ und NRZ-L.
Wenn das Digitalsignal eine 1 hat, hat der Pegel des NRZ-I-Signals einen niedrigen oder negativen Pegel, repräsentiert das Digitalsignal eine 0, dann ist der Pegel des NRZ-I-Signals High. Zwischen dem NRZ-L-Signal und dem NRZ-I-Signal besteht eine logische NOT-Funktion. Ist der eine Pegel hoch, ist der andere low und umgekehrt. Die NRZ-I-Codierung ist ebenso wie die NRZ-Codierung eine einfache Umsetzung der Pegelzustände mit Invertierung. Sie hat den Nachteil, dass bei langen 0- und 1-Sequenzen die Synchronisation außer Tritt kommen kann.

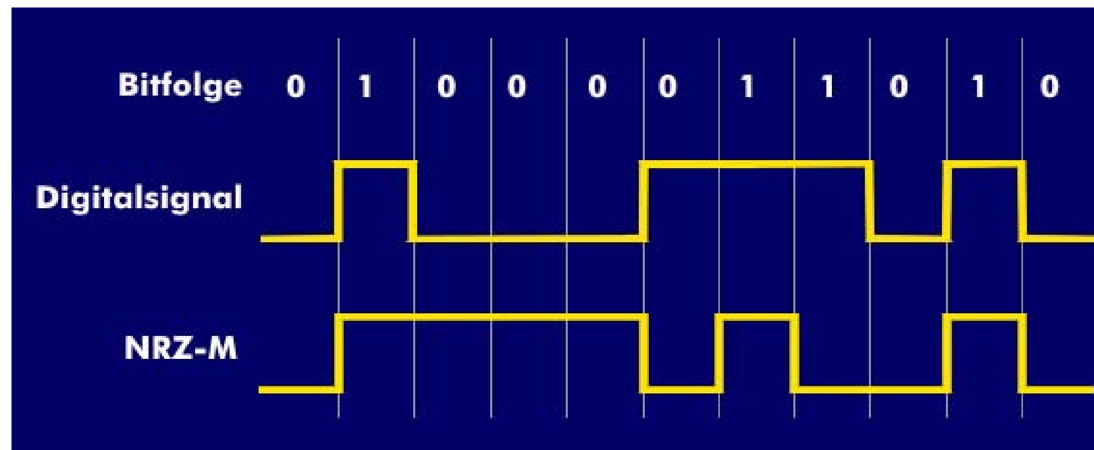



*NRZ- und NRZ-I-Codierung*

# Codierung

**NRZ-M, non return to zero mark**

Das zweiwertige *Codierverfahren NRZ*-M behebt die Nachteile der NRZ-Codierung, nämlich dass bei langen 0- und 1-Sequenzen auf der Übertragungsstrecke Gleichspannungsschwankungen auftreten und zudem die Synchronisation nicht mehr hergestellt werden kann.



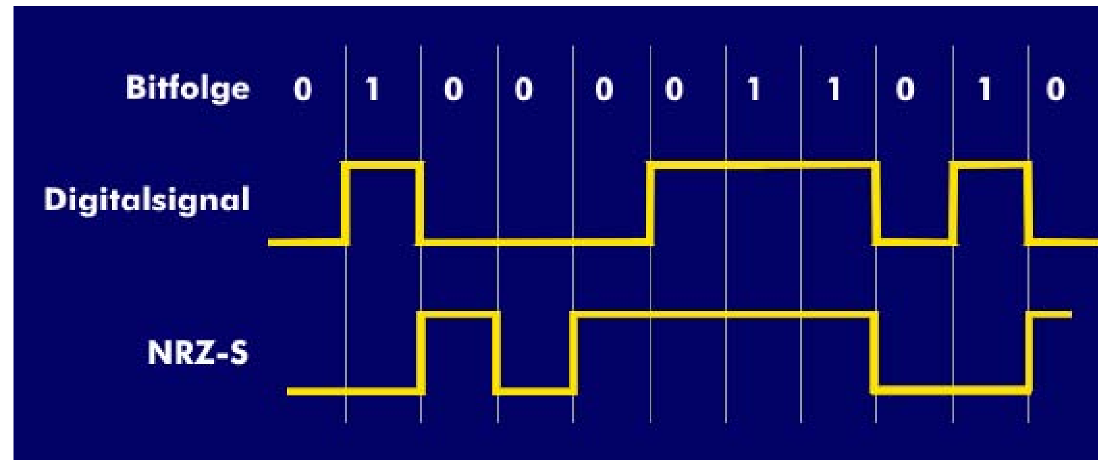

*NRZ-M-Codierung*

Bei NRZ-M erfolgt dann, wenn das Digitalsignal 1 ist, eine Zustandsänderung. Ist das Digitalsignal 1, erfolgt bei NRZ-M ein Übergang von dem vorherigen Zustand auf den anderen. War der NRZ-M-Pegel vorher High, dann wechselt er auf Low, war es vorher Low, wechselt er auf High. Hat das Digitalsignal eine 0, dann findet bei der NRZ-M-Codierung keine Pegeländerung statt.
Mit dieser Codierung werden bei langen 1-Sequenzen zusätzliche Pegeländerungen für die Synchronisation erzeugt.

**NRZ-S, non return to zero space**

Das zweiwertige *Codierverfahren NRZ*-S (Non Return to Zero Space) arbeitet im Gegensatz zu *NRZ-M* mit dem Space, wenn das Digitalsignal 0 ist.
Während das NRZ-M bei einer digitalen 1 den Pegel des NRZ-M-Signals ändert, wird bei NRZ-S das 0-Signal für die Änderung des Pegelzustands benutzt. Bei jeder 0 schaltet das NRZ-S-

# Codierung



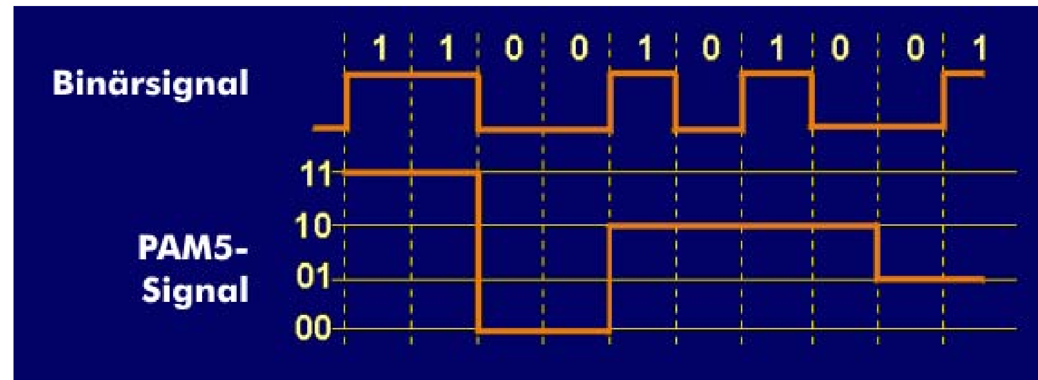

*NRZ-S-Codierung*

Signal von einem Pegel auf den anderen. Ist der Pegel vorher High, schaltet er bei einer digitalen 0 auf Low und umgekehrt. Bei einer digitalen 1 bleibt der Zustand unverändert. Mit der NRZ-S-Codierung werden bei langen 0-Sequenzen zusätzliche Pegeländerungen für die Synchronisation erzeugt.

**PAM5-Verfahren**
*PAM5 method*

Das PAM5-Verfahren ist ein *Codierverfahren*, das in Gigabit-Ethernet über TP-Kabel eingesetzt wird. Damit man Gigabit-Ethernet über Kat-5-Kabel übertragen kann, ist es erforderlich, die Übertragungsfrequenz bei der vorgesehenen Bandbreite von 250 Mbit/s unter 100 MHz zu halten.
Bei der PAM5-Codierung werden jeweils zwei Bit als Spannungswert übertragen, wie bei der Darstellung von Dibits. Die Bitkombinationen 00, 01, 10 und 11 haben bei der Übertragung vier unterschiedliche Spannungswerte und entsprechen damit vier Bitkombinationen. Der fünfte Zustand ist bei einem Übertragungspegel von null gegeben und dient bei diesem Verfahren der Fehlerkorrektur. Durch die Zweierkombination ergibt sich bei einer Datenrate von 250 Mbit/s eine Übertragungsrate von 125 MDibits/s und, da bei jedem Signalzyklus zwei Dibits übertragen werden, eine auf die Hälfte reduzierte Übertragungsfrequenz von 62,5 MHz.

# Codierung




*PAM-5-Codierung*

Da die übertragenen Signale auf den Kabeln verschliffen werden, ist die Decodierung der Dibits nicht ganz unproblematisch.
PAM5 wird in 100Base-T2 und 1000Basse-T eingesetzt.

**PAM5x5-Verfahren**
*PAM5x5 method*

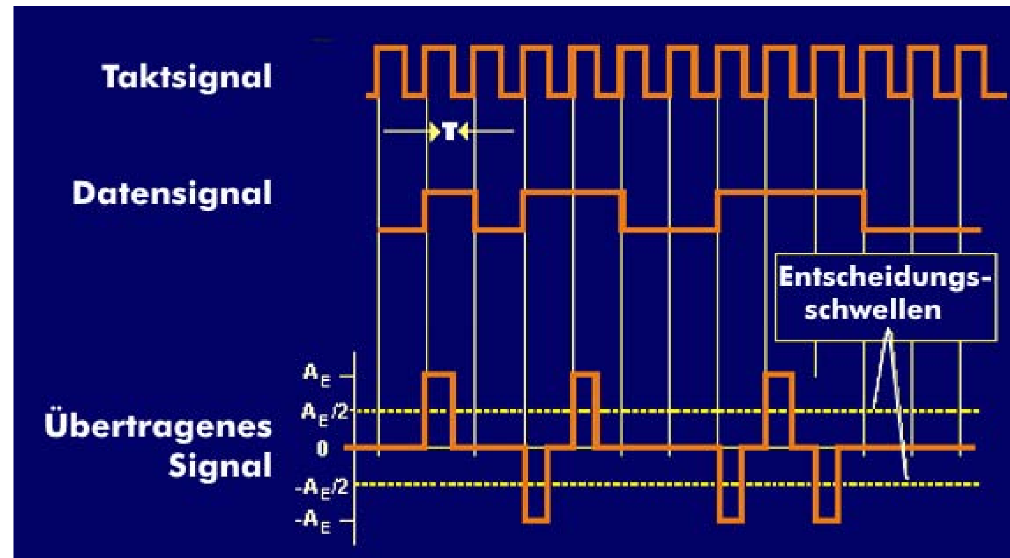
Das *Codierverfahren* PAM5x5 ist eine Pulsamplitudenmodulation, die mit fünf verschiedenen Pegeln in zwei Ebenen arbeitet. Die Pegel können die Werte -2, -1, 0, +1 und +2 annehmen, wodurch bei einer Modulation in der X- und Y-Achse insgesamt 25 verschiedene Signalpegel darstellbar sind.

**Pseudocode**
*pseudocode*

Pseudocodes werden nicht für spezielle Prozessoren geschrieben, sondern für eine Pseudomaschine. Sie werden nach dem Kompilieren in *Maschinencode* übersetzt und damit erst für Prozessoren lesbar. Da Pseudocodes nicht an einen speziellen Prozessor gekoppelt sind, können sie durch Interpreter für verschiedene Prozessoren übersetzt werden und sind daher besser auf andere Plattformen portierbar.
Die Pseudocode-Darstellung wird für die Formulierung der Algorithmen von allgemeinen Programmiersprachen in der Entwicklungsphase benutzt, da sie gut lesbar und nicht mit programmtechnischen *Codes* überfrachtet ist. Durch Pseudocodes können Programme später in die verschiedenen Programmiersprachen implementiert werden.

# Codierung

**Pseudoternärverfahren**
*pseudo ternary method*




*Pseudoternärverfahren*

Das Pseudoternärverfahren ist ein *Übertragungsverfahren* für Basisbandsysteme, das genauso arbeitet wie das *Bipolarverfahren*, bei dem der logischen »1« alternierend die Sendeamplitude »+A« und »-A« zugeordnet wird. Der Pseudoternärcode kennt drei Signalzustände: positiver Pegel, kein Pegel und negativer Pegel.
Der Unterschied zum Bipolarverfahren besteht darin, dass der Rechteckimpuls kürzer wird als die Schrittdauer. Dies hat den Vorteil, dass der Ausschwungvorgang bis zum nächsten Impuls abgeklungen ist. Das Verfahren wird in ISDN eingesetzt. In den USA wurde das Pseudoternärverfahren durch die *2B1Q-Codierung* ersetzt.

| Binäre Einheit | Anzahl der Einheiten |
|---|---|
| Bit | 1 binäre Einheit, zweiwertig |
| Dibit | 2 binäre Einheiten, vierwertig |
| Tribit | 3 binäre Einheiten, achtwertig |
| Nibble, Halbbyte | 4 binäre Einheiten, 16-wertig |
| Byte | 8 binäre Einheiten, 256-wertig |

*Binäre Einheiten und deren Wertigkeiten*

**Quadbit**
*quad bit*

Ein Quadbit oder Nibble ist eine Gruppe von vier Bit, die wie ein einzelnes Bit übertragen, verarbeitet

# Codierung

und interpretiert wird. Es repräsentiert 16 Zustände (2exp4) zwischen 0000 und 1111 und kann die Wertigkeiten des Hexadezimalsystems abbilden. Angewendet wird das Quadbit beispielsweise in der 4B/5B-*Codierung* und der QAM-Modulation zur Erhöhung der Datenrate. Werden zwei Bits zu einer Gruppe zusammengefasst, spricht man von einem Dibit, bei drei Bits von einem Tribit und bei 8 Bits von einem Byte.

**Quarternär**
*quarternary*

Um bei der Datenübertragung mehr Daten bei gleicher Datenrate übertragen zu können, arbeitet man bei den Übertragungsverfahren mit Mehrpegelverfaren. Eine Mehrpegelcodierung ist das Bipolarverfahren, das mit zwei Pegeln arbeitet, ein weiteres das Ternärverfahren, das mit drei unterschiedlichen Pegeln arbeitet, und ein drittes das Quarternärverfahren. <br>
Bei einer quarternären Codierung hat das Code-Element vier verschiedene Pegel. Die Pegel repräsentieren jeweils einen Spannungswert und können mit +3, +1, -1 und -3 gekennzeichnet sein. Jedem Pegel wird ein diskreter Digitalwert zugeordnet. Das kann ein digitaler Zustand sein, das kann aber auch ein Dibit sein, wie bei der 2B1Q-Codierung. Eine Pegeländerung entspricht damit jeweils einem Dibit.

**Quellencodierung**
*source encoding*

Bei der Quellencodierung wird das zu codierende analoge Signal in eine *binäre* Zeichenfolge umgesetzt. Bei der Digitalisierung sollte das digitale Signal die Quelleninformation möglichst nicht verfälschen. Um das zu codierende Signal möglichst effektiv übertragen zu können, kann es für die Bandbreiteneffizienz vor der Übertragung so komprimiert werden.
Beispiele für die Quellencodierung sind die Pulscodemodulation und die Deltamodulation.

**Quibinärcode**
*quibinary code*

Der Quibinärcode ist ein *7-Bit-Code*, in dessen 7 Bit immer nur zwei „Einsen“ vorkommen. Es sind also nie mehr als zwei Bit auf „1“ gesetzt.

# Codierung

| Dezimal | Quibinär | Dezimal | Quibinär |
|---|---|---|---|
| 0 | 0000101 | 5 | 0010010 |
| 1 | 0000110 | 6 | 0100001 |
| 2 | 0001001 | 7 | 0100010 |
| 3 | 0001010 | 8 | 1000001 |
| 4 | 0010001 | 9 | 1000010 |

*Dezimalwerte im Quibinärcode*

Vom Aufbau her verschiebt sich das erste Bit bei jeder zweiten Ziffer um eine Bitstelle nach vorne, während das zweite Bit immer durch ein Least Significant Bit (LSB) oder durch das davor liegende Bit dargestellt wird. Der Quibinärcode ist redundant und fehlersicher. Er wird allerdings kaum eingesetzt, weil er mit seinen 7 Bit nicht in das Byte-Schema (8 Bit) passt.

**READ, relative element address destinate**

Relative Element Address Destinate (READ) ist ein zweidimensionales *Codierverfahren*, das bei der Faxübertragung nach Gruppe 3 eingesetzt wird. Bei dieser *Lauflängencodierung* wird die Differenz zwischen einer Referenzzeile und der jeweils abgetasteten Zeile übertragen. Die Referenzzeile wird gespeichert und nach einer bestimmten Anzahl von Zeilen (K-Faktor) gegen eine neue Referenzzeile ausgetauscht. Das übertragene Signal ist also abhängig vom Inhalt der momentan abgetasteten Zeile und der Referenzzeile.

**Reed-Solomon-Code**
*RS, reed solomon code*

Der Reed-Solomon-*Code* (RS) ist ein Fehlerkorrekturverfahren, das in der Mobilkommunikation bei CDPD, in der drahtgebundenen Kommunikation bei ADSL und ATM, aber auch bei der Datenspeicherung auf Compact Discs (*CD*) und DVDs eingesetzt wird.
Beim Reed-Solomon-Code handelt es sich um ein Verfahren mit Vorwärts-Fehlerkorrektur (FEC), das die Korrektur fehlerhaft übertragener Datenblöcke ermöglicht. Die FEC-Daten werden in einem Code-Wort in einem Datenblock den Nutzdaten angehängt und mit diesen zusammen übertragen. Das *Codewort* besteht aus den Nutzdaten und einer Anzahl an Reed-

# Codierung

Solomon-Prüfbytes sowie einem oder mehreren DMT-Symbolen. Wurde der RS-Code mit 255/239 gewählt, dann bedeutet das, dass je 239 Datenbytes zusätzlich 16 Bytes für die Vorwärts-Fehlerkorrektur hinzugefügt werden.
Die mathematische Beschreibung ähnelt der des *CRC*-Verfahrens, wobei eine vorher gebildete Reed-Solomon-Prüfziffer für die Polynomdivision genutzt wird.
Für den Einsatz in Compact Discs (CD) wurde das Verfahren abgewandelt in das CIRC-Verfahren.

**RLE, run length encoding**

Die Run Length *Encoding* (RLE) ist eine verlustfreie Kompression, bei der eine Folge von gleichen Zahlen, Zeichen oder Buchstaben durch ein einziges Symbol und die Angabe über die Anzahl der gleichen Symbole ersetzt werden. Man spricht in diesem Zusammenhang von einem Run, das ist eine Abfolge von identischen Zeichen, und der Länge der jeweiligen Sequenz, die in einem Run Counter gespeichert wird.
Die Lauflängencodierung beseitigt Redundanzen in Form von Wiederholungen und eignet sich besonders für Grafiken und Bilddateien mit wenigen Farben. In der Computergrafik wird das RLE-Verfahren bei speicherintensiven Rastergrafiken angewendet und ist dann am effizientesten, wenn es sich um einfache Grafiken mit wenig Farben und großen Farbflächen handelt. Um die Effizienz des Verfahrens zu erhöhen, kann der Algorithmus eine Wegoptimierung vornehmen durch die Wahl des effektivsten Laufwegs, der zeilensequenziell, im Zick-Zack oder auch mäanderförmig sein kann.
So wird bei Run Length Encoding beispielsweise die Zahlenfolge aaaaaa durch 2 Bytes ersetzt: ein Byte für den Buchstaben „a“, das andere für die Anzahl 6. Liegen die Buchstaben, Ziffern und Zeichen in binärer Form vor, so kann die Lauflängencodierung noch effizienter arbeiten.

**Buchstabenfolge**
aaaaabbbccccccccddeeee
Länge: 24
5a3b8cdd4e
Länge: 10

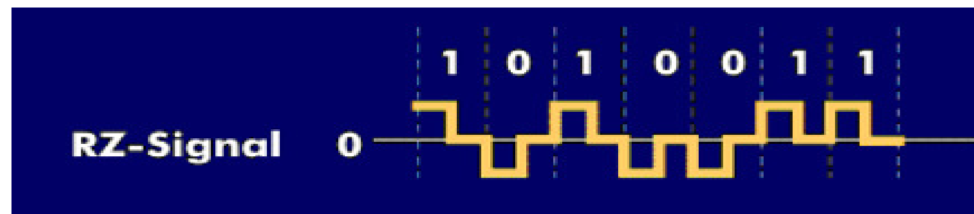
**Binärfolge**
000000000011110000011111111
Länge: 27
10 4 5 8 = 1010 0100 0101 1000
Länge: 16

*Lauflängencodierung am Beispiel von Buchstaben und Binärdaten*

Die Lauflängencodierung zeichnet sich durch ihre Einfachheit und Geschwindigkeit aus. Bei der Kompression werden gleiche Werte so lange eingelesen, bis sich der Wert ändert. Wert und die Anzahl der gleichen Werte wird festgehalten. Bei der Dekompression wird lediglich der Wert ausgelesen und die entsprechende Anzahl an Bytes ausgegeben.
RLE wird bei verschiedenen Grafikdateiformaten wie TIFF, dem Bitmap- und dem TGA-Dateiformat eingesetzt, aber auch bei Faxübertragungen.

**RZ, return to zero code**
*RZ-Codierung*

Der Return to Zero *Code* (RZ) ist ein Nachrichtenübertragungscode, der im Wesentlichen der *NRZ-Codierung* entspricht, mit dem Unterschied, dass das High-Signal nach dem halben Bitintervall wieder auf 0 zurückgeht.
Das Low-Signal wird nicht codiert. Die Taktsynchronisation des Empfängers kann durch die Umschaltflanke des codierten High-Signals erfolgen, ist aber bei langen Bitfolgen nicht sicher.



*RZ-Verfahren*

Bei langen 0- oder 1-Sequenzen kann sich außerdem der Gleichspannungsanteil verändern.

# Codierung

**RZ-I, return to zero inverted**
*RZ-I-Codierung*

Die *RZ-I-Codierung* (RZ-I) ist die invertierte Version der RZ-Codierung. Bei der RZ-I-Codierung sind die logischen Zustände für die digitale 0 und 1 umgekehrt als bei der RZ-Codierung. Die Synchronisation des Taktimpulses des Empfängers erfolgt durch die Umschaltflanke des codierten High-Signals.

**SBC, subband coding**
*SBC-Codierung*

Das *Subband Coding* (SBC) ist ein *Codierverfahren*, das speziell für das Bluetooth-Profile A2DP entwickelt wurde. Die SBC-Kompression erzeugt Bitströme von 127 kbit/s bis 345 kbit/s und stellt niedrigere Anforderungen an die Hardware als beispielsweise MP3. Aus den verschiedenen Bitraten resultieren unterschiedliche Qualitätsmerkmale für Mono und Joint Stereo.
Die SBC-Sender und -Empfänger verständigen sich vor der Übertragung auf eine Abtastfrequenz und Bitrate. Den Time Stamp für den richtigen Wiedergabepunkt leitet der Sender aus der Abtastrate ab.

**Scrambler**
*SCR, scrambler*

Ein Scrambler ist ein Pseudozufallsgenerator, mit dem *binäre* Signale quasi-zufallsbedingt verwürfelt werden. Es handelt sich dabei um ein mehrstufiges Schieberegister, bei dem die Ausgangssignale der einzelnen Kippstufen logisch miteinander verknüpft und auf den Eingang des Scramblers rückgekoppelt werden. Die Quasi-Zufallsfolge ergibt sich aus der Verknüpfung und der Anzahl der Kippstufen des Schieberegisters. Da die Pseudozufallsfunktion bekannt ist, kann das Digitalsignal nach der Übertragung in einem Descrambler wieder in das Originalsignal rückgewandelt werden.
Ein Scrambler vermeidet gleichbleibende Signalmuster wie sie beispielsweise im Idle-State auftreten und sorgt für eine bessere Verteilung der Signalenergie über den Frequenzbereich. Der Scrambler macht aus jedem Signal eine Zufallsfolge.

# Codierung

Scrambler werden in den meisten Modems der V-Empfehlungen eingesetzt, so u.a. in V.28, V.29, V.32, V.34 und V.42.

**Scrambling**
*scrambling*

Beim Scrambling wird ein Datenstrom nach einem festgelegten mathematischen Polynom verwürfelt. Die Verwürfelung wird im Prinzip durch einen Pseudozufallsgenerator mit der Länge 2expn-1 mit n-Rückkopplungsstufen ausgeführt. Diese Technik dient der Vermeidung gleich bleibender Signalmuster und sorgt für eine gleichmäßigere Verteilung der Signalenergie über den gesamten Frequenzbereich. Durch das Scrambling wird eine Konzentration der Signalenergie auf ein schmales Frequenzband mit hoher Amplitude verhindert, wie es bei langen sich wiederholenden Bitmustern wie beispielsweise beim Idle-Signal ohne Scrambling auftreten würde.
Durch das Scrambling können nicht codetransparente *Übertragungsverfahren* in ihren Eigenschaften verbessert werden. *Scrambler* werden in der Modemtechnik, der Basisbandtechnik und bei der Übertragung von Rundfunk- und Fernsehsignalen eingesetzt. Die meisten Modems der V-Serie nutzen die Scrambling-Technik. In lokalen Netzen wird diese Technik in Transceivern angewendet.

**STBC, space time block coding**
*STBC-Codierung*

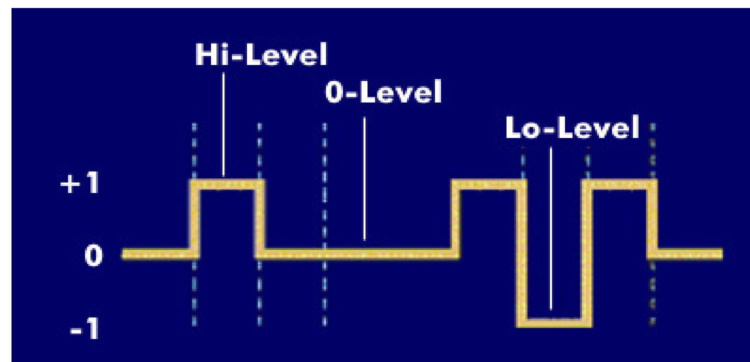
Die Raum-Zeit Block *Codierung* (STBC) ist ein *Übertragungsverfahren* von Funknetzen, bei dem die zeitliche, die spektrale und auch die räumliche Dimension der Übertragungsstrecke ausgenutzt werden.
Bei der STBC-Codierung wird der Datenstrom zur Übertragung in Datenblöcke codiert. Diese werden über räumlich verteilte Antennen zeitlich verteilt übertragen. Bei dem Verfahren werden senderseitig Antennen-Arrays eingesetzt, empfangsseitig sind keine erforderlich, allerdings trägt ein empfangsseitiges Antennen-Array zur Verbesserung der

# Codierung

Empfangsbedingungen bei. Dieses Übertragungsverfahren, bei dem vielfältige Kopien der Daten empfangen werden, ist auch bekannt als Raumdiversität und ist die Strategie des MIMO-Verfahrens.
Die STBC-Codierung wird gewöhnlich als Matrix dargestellt. Jede Reihe repräsentiert einen Zeitschlitz und jede Spalte die Übertragung von einer Antenne. In dieser Notation stellt „S" das modulierte Symbol dar, das im Zeitschlitz „y" von der Antenne „x" übertragen wird.
Beim Space Time Block Coding (STBC) unterscheidet man zwischen orthogonalen Raum-Zeit-Block-*Codes* (OSTBC) und nicht-orthogonalen Raum-Zeit-Block-Codes (NOSTBC). Zu letzteren gehört QSTBC (Quasi-orthogonal). Mit OSTBC wird bei MIMO mit zwei Antennen die volle Datenrate und Diversität erzielt. Haben MIMO-Systeme mehr als zwei Antennen, dann kann die volle Datenrate nur mit QSTBC erzielt werden, allerdings mit eingeschränkter Diversität. Die STBC-Codierung wird u.a. vom Enhanced Wireless Consortium (EWC) für 802.11n vorgeschlagen.

**Ternär**
*ternary*

Bei den Übertragungsverfahren versucht man durch Mehrpegelverfahren die Datentransferrate bei gleicher Datenrate zu erhöhen. Dazu werden den einzelnen Pegeln jeweils ein Bit zugeordnet, wie beim Bipolarverfahren, oder ein Dibit wie beim Ternärverfahren oder ein Quadbit wie beim Quarternärverfahren.
Beim Ternärverfahren basieren die Signalelemente oder Code-Elemente auf drei diskreten Werten: auf einem positiven



*Ternärsignal mit den drei Pegeln*

# Codierung

Hi-Level, dem 0-Level und einem negativen Lo-Level.
Jedem dieser Pegel kann eine Bitkombination zugeornet werden. Beim Ternärsystem unterscheidet man in der Darstellung zwischen dem gewöhnlichen Ternärsystem und dem balancierten. Während das gewöhnliche Ternärsystem die drei Pegelwerte mit den Ziffern 0, 1 und 2 darstellt, wird beim balancierten Ternärsystem mit Vorzeichen gearbeitet.
Der Nullpegel entspricht der „0", der positive Pegelwert der „1" und der negative der „-1". Da die Schreibweise zu Irrtümern führen kann, wird anstelle des Vorzeichens die „-1" auch als Eins mit Unterstrich dargestellt, oder es wird in einer Fußnote auf die Balanced-Version hingewiesen.

**Trellis-Codierung**
*TC, trellis encoding*

Die Trellis-*Codierung* ist ein Verfahren mit so genannter Vorwärts-Fehlerkorrektur (FEC) das in verschiedenen Hochgeschwindigkeits-Modems nach V.32, V.32bis, V.32terbo, V.34 und V.fast eingesetzt wird. Bei dem Trellis-Verfahren werden zu sichernde Daten sendeseitig mit einem Sicherungspolynom verknüpft; es werden also ständig Redundanzbits in den Datenstrom eingefügt. So werden bei einer Trellis-Codierung mit vier Bit die vier Datenbits in fünf Bits codiert und übertragen, wobei das fünfte Bit aus den vier Datenbits errechnet wird und das Sicherungspolynom darstellt. Dieses Bit dient Redundanzzwecken und verknüpft die zeitliche Reihenfolge der übertragenen Datenbits.
Die Signalübergänge sind dadurch nicht mehr frei, sondern unterliegen einer gewissen Gesetzmäßigkeit, mit deren Hilfe empfangsseitig fehlerbehaftete Signalfolgen ausgefiltert werden. Das Trellis-Verfahren erzielt gegenüber der reinen Quadraturmodulation einen um 3 dB besseren Störspannungsabstand. Die Trellis-Codierung wird auch bei ISDN verwendet.

# Codierung

**Übertragungsverfahren**
*encoding*

Unter Übertragungsverfahren - in der angelsächsischen Literatur spricht man von *Codierung* - versteht man Methoden zur Übertragung von Bitströmen. Wichtige Anforderungen an die Übertragungsverfahren sind die Gleichspannungsfreiheit, der Bandbreitenbedarf und die Möglichkeit der Taktrückgewinnung. Die unterschiedlichen Übertragungsverfahren unterscheiden sich darin, dass der zu übertragende Bitstrom im Pegel der zu codierenden Impulse, in deren Phasenlage, der Zeitdauer oder den Flanken Veränderungen erfährt. Diese werden empfangsseitig zur eindeutigen Bitstromerkennung und somit auch zur Taktrückgewinnung genutzt.
Man unterscheidet nach der Übertragungsart in Basisband-Übertragungsverfahren und Breitband-Übertragungsverfahren. Erstere sind unmoduliert, Letztere mit Modulationsfrequenzen unterlegt. Die wichtigsten Basisband-Übertragungsverfahren sind das *Einfachstromverfahren*, das *Doppelstromverfahren*, das Ternärverfahren mit der *MLT-Codierung*, *8B6T-Codierung*, *CAP-Codierung* und weiteren, das *Pseudoternärverfahren* mit der *AMI-Codierung*, die *Manchester-Codierung*, das *Bipolarverfahren* mit der *HDB-Codierung* und das *Coded-Diphase-Verfahren*.
Zu den bekanntesten Breitbandübertragungsverfahren gehören die diversen Pulsmodulationen wie z.B. Pulsamplitudenmodulation (PAM), Pulscodemodulation (PCM), Pulsweitenmodulation (PWM) und Umtastverfahren wie PSK, FSK, ASK, DPSK.
In der Funk- und Mobilfunktechnik haben sich Mehrträgerverfahren wie Frequenzmultiplex etabliert. Bei manchen Verfahren können gleichzeitig mehrere Unterträger auf mehrere Subkanäle für verschiedene, parallel bediente Mobilstationen verteilt werden. FDMA und das mit orthogonalen Unterträger arbeitendes OFDMA sind solche Verfahren.

# Codierung

**Viterbi-Code**
*Viterbi code*

In den 60er Jahren hat Viterbi einen nach ihm benannten Decodieralgorithmus veröffentlicht, der auch heute noch in *Faltungscodes* eingesetzt wird. Er basiert auf einer Schätzung der Empfangsfolge. Dabei wählt der Viterbi-Algorithmus aus allen möglichen Codefolgen diejenige Empfangscodefolge aus, die die geringste Hamming-Distanz hat. Es handelt sich dabei um die Codefolge, die sich nur geringfügig von der empfangenen Folge unterscheiden. Für diese Bewertung wurde eine Metrik eingeführt, mit der die Empfangscodefolge auf Bitstellen hin untersucht wird.
Mit dem Viterbi-*Code* können Bitfehler korrigiert werden.

# Impressum

ITWissen
Das große Online-Lexikon
für Informationstechnologie

**Herausgeber**

Klaus Lipinski
Datacom-Buchverlag GmbH
84378 Dietersburg

ISBN: 978-3-89238-181-5

**Codierung**



Trotz sorgfältiger Recherche wird für die angegebenen Informationen keine Haftung übernommen.






Layout & Gestaltung: Sebastian Schreiber
Titelbild: © traff - Fotolia.com
Produktion: www.media-schmid.de

Weitere Informationen unter **www.itwissen.info**